jrg:

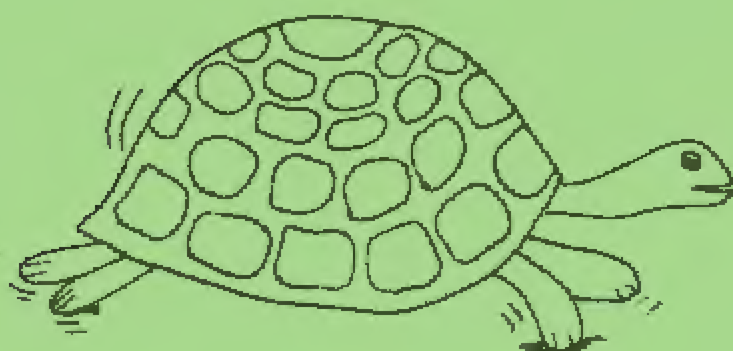
nr:

NOVEMBER 1983

++

- ACORN Nieuws -

++



6000 BD

WIE BIEDT

ER MEER ? ?

9600 BD!

TURTLE GRAPHICS

++

ACORN NIEUWS Uitg.Fed.Acorn Computerclubs Ned/B.

Verschijnt 6-8 x p/jr.

INHOUD

In deze uitgave weer voor elk wat wils.
We hebben nu een ruime keus gekregen uit snelheden van ons COS. 300, 1200, 3000, 6000 en 9600 Baud.
Nog even verder en het hele geheugen staat al op de band voordat het commando gegeven is.

Even wat over de twee artikelen over een tweede 16K RAM-kaart.
Tegelijkertijd, doch geheel onafhankelijk van elkaar werd er gewerkt aan een tweede 16K-kaart. Het resultaat is hetzelfde, alleen de uitgangspunten verschilden.
Gerhard Visser ging uit van een zo simpel mogelijke oplossing, zonder gebruik te maken van een onderkaart, terwijl Ap Peters er van uit ging om het stroomverbruik en het aantal onderdelen zo gering mogelijk te houden.
Beide kaarten werken perfect!
Aan u de keus om een oplossing te kiezen die voor u het meest aantrekkelijk is.
Succes.

CASSETTEBANDJES

De algemeen gebruikelijke manier van opslag voor de programma's van onze computer is de compactcassette. Nu zijn er cassettes en cassettes. Via uitwisseling van programma's (met de oude en vertrouwde 300 baud) heb ik verschillende soorten en merken cassettes onder handen gehad. Vele van de problemen die bij het laden van de bandjes optraden zijn te wijten aan de kwaliteit van die bandjes.
Vandaar het volgende relaas, overgenomen uit de Consumentengids van september 1983, met daaraan verbonden een klein verhaaltje mijnerzijds.
Omdat er vele goede en goedkope bandjes in de categorie ferro-bandjes zijn, is het verhaaltje daartoe beperkt.

Doordrukdemping.
De cassetteband wordt bij een opname gemagnetiseerd door de kop van de recorder. Dit gemagnetiseerde band kan zelf weer magnetiseren, bijvoorbeeld cassetteband dat eronder of erboven wordt opgewikkeld. Als dat gebeurd, is dat te horen als een voor- of na-echo van het opgenomen geluid.
Een band met weinig doordrukdemping heeft dan ook onze voorkeur, omdat de Atom blokgolven met een grote echo op de achtergrond soms echt niet begrijpt.

Drop-outs, stabiliteit.
Om een goede, constante en gelijkmatige weergave te krijgen, moet de magnetische laag van de band zeer gelijkmatig zijn, en de band zelf (de "drager") soepel en niet-rekkend. Anders kan het geluid soms wegvallen (drop-outs), over bepaalde afstanden varieren of instabiel zijn.
De Acornist zal zich wel hoeden voor het wegvallen van net het meest spectaculaire spelletje of de ingewikkelste dynamische simulatieprogramma's.

Afspeelverlies.
Wordt het signaal zwakker of instabieler elke keer dat je een bandje afspeelt? Enkele bandjes konden vijfentwintig keer afspelen al moeilijk lijden.

Enkele bandjes die voor ons, het voorafgaande in ogenschouw nemende, het meest in aanmerking komen zijn:
BASF LH EXTRA I, SONY BHF, TDK D, TDK AD, TDK AD X.

Deze test had betrekking op C90 bandjes, maar de door ons vaak gebruikte C60 bandjes zijn van een vrijwel gelijke kwaliteit. Voor ruim onder de f1.3,-- zijn dus al goede en probleemloze bandjes te verkrijgen.

Voor de volledige test zie de Consumentengids september 1983.

Gerhard Visser.

ZAKELIJKE ZAKEN

voorzitter: A. Millenaar
Thorbeckestraat 38
8161 DP Epe

secretaris: vacant
tijdelijk correspondentieadres:
FACC
Haringvliet 391
8032 HL Zwolle

penningmeester: G. Visser
Haringvliet 391
8032 HL Zwolle
tel: 038-546561

redacteur Acorn Nieuws: A. Otten
J.A. de Gravelaan 17
2381 TA Zoeterwoude
tel: 01715-1929

postgiro: 5244293 bankgiro: 93.32.87.283
Beide tnv penningmeester Acorn Computerclub te Zwolle.

De "clubwinkel"

Geheugenkaart: 16 kByte extra in de Atom	fl. 35.--
Schakelkaart: meerdere EPROM'S op Axxx (incl. 74LS133)	fl. 45,--
Programmerkaart: zelf programmeren van EPROM'S	fl. 17.--
N I E U W speciale herdruk Acorn Nieuws jaargang 1982	fl. 6,--

(Bij het bestellen van de schakelkaart kan uw geduld mogelijk even op de proef gesteld worden. De voorraad is ni. uitgeput en met het aan laten maken van een nieuwe serie wachten we even. Dit om de seriegrootte af te stemmen op de behoefte.)

Bovenstaande artikelen kunt u bestellen door het desbetreffende bedrag over te maken op een van bovenstaande gironummers. U kunt natuurlijk ook de bestelling via de regionale penningmeesters laten lopen, zodat u de bestelling veelal tijdens uw volgende bijeenkomst in ontvangst kan nemen.

Archiefdiensten

Voor bandjes, datasheets, EPROM's en alle andere vormen van hulpverlening kunt u terecht bij de regionale archiefdiensten. Mocht in uw regio iets ontbreken, dan kan het regionaal bestuur u vast wel verder helpen.
De twee landelijke archiefdiensten geven regelmatig een overzicht uit van wat er in die archieven aanwezig is. De regionale archiefdiensten kunnen daaruit dan een keus maken van wat zij in het archief willen hebben.

Het bestuur deelt mede

De oprichter van de Acorn Computerclub, Gerard Borghaerts, heeft onlangs zijn functie van secretaris in het bestuur van de Federatie neergelegd. Wij zijn hem veel dank verschuldigd voor de vele werkzaamheden die hij verricht heeft.
Inmiddels is de procedure om een nieuwe secretaris te kiezen van start gegaan.

Overbelasting

Het komt regelmatig voor dat de leden in geval van problemen (hardwarezaken, softwarezaken, verkrijgbaarheid van bandjes etc.) te rade gaan bij het hoofdbestuur of bij bijvoorbeeld de auteur van het desbetreffende artikel in Acorn Nieuws. Voor deze mensen kan het op deze manier wel erg druk worden. Er is een voorbeeld bekend van meer dan 100 reacties op een artikel in Acorn Nieuws in een paar dagen tijd.
Om te voorkomen dat deze mensen daardoor geen tijd meer overhouden om hun eigenlijke hobby uit te oefenen, verzoeken wij u zeer dringend om voorkomende vragen eerst zoveel mogelijk binnen uw eigen regio op te lossen (tussen twee haakjes: Zelf problemen oplossen, al dan niet met hulp van enkele regio-experts, is veel leerzamer en misschien nog leuk ook).

Contributie 1984

Bij deze Acorn Nieuws ingesloten treft u een acceptgirokaart aan. Deze kunt u gebruiken om uw contributie voor 1984 over te maken.
Om het de fed. penningmeester een beetje te vergemakkelijken wordt u verzocht op de desbetreffende plaats op die acceptgirokaart de regio in te vullen waarvan u lid bent.
Van de fl. 50,-- die u stort blijft weer een gedeelte in de federatieve kas zitten. Hiervan wordt dan o.a. het Acorn Nieuws vervaardigd. Het resterende bedrag (fl. 15,--) wordt doorgestuurd naar de regio's. Daar worden dan eventuele regioblaadjes en bijeenkomsten van bekostigd.
U wordt verzocht niet al te lang te wachten met het overmaken van de contributie, zodat dit al in een vroeg stadium aan de regio's doorgegeven kan worden.

Herdruk Acorn Nieuws 1982

Eindelijk is het dan zover. De uitgaven van Acorn Nieuws jaargang 1982 zijn herdrukt en in een uitgave samengebundeld.
De uitgave komt uit in dezelfde vorm als die van het boekje dat u momenteel aan het lezen bent en is ca. 100 pagina's dik. Alle artikelen zijn rubrieksgewijs geordend. Programma's zijn niet opgenomen, omdat de uitgave dan veel te groot zou worden. Via de regionale bandjesarchieven is echter een bandje verkrijgbaar waar al deze programma's opstaan. Zo hoeft u ze zelf niet meer in te typen.
De "nieuwe" leden die een groot stuk extra informatie willen hebben en de "oude" leden die hun eigen stukgelezen exemplaren willen vervangen kunnen het bestellen zoals elders in uw lijfblad vermeld staat. Wie reeds binnen zijn eigen regio het boekwerkje besteld heeft, kan het daar natuurlijk verkrijgen.
Nog even dit: De drukker begint aan de herdruk 1982 zodra deze Acorn Nieuws de deur uit is. Wanneer u supersnel reageert met bestellen, kan het zijn dat u een weekje geduld moet hebben.

Uit de regionale clubbladen:

Acorntjesbrood 2
NEW ATOM COMMANDS (uit Personal Computer World): Door het intypen van een speciaal karakter (^) maakt u het scherm schoon. / BREAK IS UIT - NMI IS IN: Door een NMI te geven (bijvoorbeeld met een apart schakelaartje) wordt uw programma heel netjes onderbroken. Daarna is een simpele druk op een toets genoeg om de draad weer op te pakken. / GEHEUGEN TEST-PROGRAMMA: Test het RAM-geheugen op een uiterst degelijke wijze. / WORD-PACK UTILITY: Stelt u in staat om de Wordpack uitvoer geformateerd op het scherm te laten verschijnen, zonder dat de printer telkens moet "meelopen". / ZERO-PAGE REFERENCE SEARCH: Zoekt alle door een assemblerprogramma gebruikte zero-page adressen bij elkaar. / WRITE-PROTECT OP DE SCHAKELKAART: Zo doe je dat! / WAAR OF NIET WAAR? Over logische operators en hoe je die gebruikt. / INTEGER-DECIMALEN (bewerkt uit PCW): Cijfers achter de komma, ook zonder FP-rom.

Het bronsgroen eikeltje 1
VIJFVOUDIG SCHERM: Vijf verschillende beeldscherm-geheugens, snel tevoorschijn gehaald door een bepaalde toetsaanslag. / CATALOG: Geeft een overzicht van de aanwezige programmatitels, mits die in de eerste REM-regel van elk programma staan. Met het startadres van de betreffende TEXT-ruimten. / HEXDUMP WITH ASCII: Uit de Willow-softtool. / ROMAR-BOX: Een hulpmiddel bij het ontwikkelen van software. De box bevat een twintigtal commando's voor het maken, testen en op band zetten van programma's. Een complete beschrijving, van Arie Marchal en Ed Ronda. / SCREENCOPY: Een programma om een mode 4 scherm te laten uitprinten. Voor de printertypes MX82 en GP250. / MATRIX: Een programma om twee matrices met elkaar te vermenigvuldigen. / KALENDER: Voor elk jaar tussen 1583 en 2400, afgedrukt door een Seikosha GP250A printer.. / SCHEMA van een goede voeding voor onze ATOM.

Het bronsgroen eikeltje 2
SOURCEMAKER: Eerst keken we gewoon naar de geheugenplaats, toen kwam als hulp een disassembler en nu hebben we de Source-maker. Marchal en Ronda hebben weer hun best gedaan. / COMPLEX REKENEN: Werkelijkheid en denkbeeldigheid bij het rekenen, onze Atom zorgt voor de oplossing. / WERELDTIJD: Het tijdsverschil rekent de Acorn voor u uit. / DE 6847 en zijn (on)mogelijkheden. Een artikel uit Practical Electronics. / UTILITY'S: Drie routines die u op kunt nemen in een eigen "toolkit" of in de "schakelsoft", n.l. Pauze, Tabulatie en Clear Screen onder de cursor. / SCHAKELKAART: Over de battery back-up, een schema en vele wetenswaardigheden.

Atomix 4
Deze uitgave bevat, behalve enkele huishoudelijke pagina's, de programma's DUP en INDEX, beide opgenomen in dit nummer van Acorn Nieuws.

De Cursor 5 en 5a
GRAPHISCHE GRAPJES: "Spreken" voor zich. / TOETSEN BIEPER: Geeft elke toets z'n eigen piepje. / CONWAY TRANSFORMATIE: Is dit niet zoiets als "LIFE"? / NUMERIEK TOETSENBORD: Kennelijk afkomstig van een pocket calculator, aangesloten op de Atom-VIA. Met bijbehorend programma. / SPACE SHUTTLE: Zonder kleerscheuren vrachtjes afleveren in de ruimte. / CHARON: (u weet wel: de functietoetsen

van Frans van Hoesel) aangepast voor het gebruik in "STATM.UITBR.JH" (van archiefbandje 6). door de breakvector niet te gebruiken. / 3600 BAUD: Zie in dit nummer van Acorn Nieuws.

STALLET september 1983
BASIC-INTERPRETER: Volledig voor u uitgeplozen tot een keurige "source-listing" met overvloedig commentaar. / ASSEMBLER TRUC: Om bij assembleren met afgezet scherm (P.$21) toch een melding te krijgen dat er een te grote sprong is gedetecteerd. / PI: Hoe nauwkeurig is de Atom-pi? Nauwkeuriger dan u denkt. / CASSETTE TRUC: Twee stukken programma in een keer laden, als ze door middel van deze truc geSAVEd zijn. / SCHEMA van de club-ramkaart. / TOONKUNST: Beter geluid uit uw VIA door een 4-bits D/A-converter en voorgeprogrammeerde golfvorm (jaja, eindelijk iemand die mijn artikel gelezen heeft!). / HET BENADEREN van functies door veeltermen: als u deze titel niet begrijpt, leest u dan maar door. / REKENEN MET BREUKEN: Een oefenprogramma. / BIT- OF SHAPE-ANALYSE: Geeft de inhoud van de aangegeven geheugenplaatsen weer in eenen en nullen, dus binair, met daarnaast als hex-getal. / LADEN VAN TWEE GEHEUGENBLOKKEN in een keer, net als de "cassette truc" hierboven, maar met toelichting. / KRAKEN EN SAVEN: Om een beveiligd programma van 300 naar 1200 Baud te kunnen versnellen.

DATA CHECK 4
TWEEDE VIDEO PAGE: Gestapeld op de bestaande video-RAM's en naar wens uitschakelbaar, als geheugensteuntje. / USING ATOM WITH BBC BASIC: (Engelstalig) verslag van een bezitter van een naar BBC-Basic geconverteerde Atom. De conclusie is dat men voor spelletjes en graphics beter bij de Atom kan blijven (uit Personal Computing Today okt. '82).

DATA CHECK 5
KNIPPERROUTINE: Een verbeterde versie van A.N.4. / SELBSTBAUPLOTTER: Een (duitstalig) artikel over het zelf in elkaar knutselen van een X-Y-plotter (uit MC dd 8/1983). / FLOPPY DISC CONTROLLER in combinatie met de 64K RAMkaart van ECD: Een sterk vereenvoudigde floppy-disc controller, gebaseerd op het ACORN-DOS.

ACORN NEWSLETTER 4
VOEDINGEN: Voor de elektronische leek een uitstekend verhaal, waarin alles over gestabiliseerde voedingsapparaten wordt uitgelegd. / ATOM BASIC COMPILER: Van de Stichting Electronica Workshop in Delft een beschrijving van haar BASIC compiler, die ons in staat stelt BASIC-programma's maximaal 10x sneller te laten "draaien". / BOEK- EN TIJDSCHRIFTBESPREKINGEN / PROGRAMMEERSCHEMA: Hoe zet ik een programma op, in 8 stappen. / SCROLL-programmaatje, dat het gehele video-geheugen scrollt, dus ook in hogere modes bruikbaar. / MAKING A HABIT OF THE HOBBIT: Een engelstalig verslag over de zegeningen van een "fast floppy tape system": d.w.z. een automatische micro-cassetterecorder (uit Acorn User). / DERDE GRAADSVERGELIJKING: Een oplossings programma met uitgebreide toelichting. / DE ACORN ATOM. EEN GEBRUIKERSRAPPORT uit Personal Computer & Software (in het nederlands). Vol lof, uiteraard.

HET BANDJESARCHIEF
=================

Hier is dan het beloofde artikel over de uitbreiding van het bandjesarchief.
Ik ben nu bezig met in ieder geval 3 bandjes.
Van deze 3 is (nu ik dit stuk schrijf) nr. 11 al klaar en 12 & 13 zijn in de maak.Deze 3 banden zijn door de regio's bij mij te bestellen.Dit laatste doe ik omdat het anders te lang zou duren voordat alle regio's het bandje, via de rondstuur methode, hebben ontvangen.Deze methode gebruik ik alleen nog maar om programma's te laten becommentarien.Ik zorg ervoor dat ik een kleine voorraad heb liggen.dus heren regio-archievarissen stuur uw cassette met retour porto maar op.

En dan nu de inhoud van de bandjes:

BAND 11 :

Kant A :

VISLOAD , GDUMP-TOORMAN , GP-80 H.R. , GRAF-RITE , SRT9 A.M. , PLOTTER , BINGO , DEMO-JOY , 3-D PLOT , BASIC UITBR. , GRAFIEKEN , JUMPERS , AFRONDEN , AFRONDEN , MUSIC , HYPOTHEEK , HOOFDREKENEN , JOTTING , JOYSTICK , MUZIEKDOOS , WILHELMUS , PRIEMGETALLEN.

Kant B :

LOTTO , NIMSPEL , BUBBLESORT , AFSTANDSBER. , FIND , TOOLBOX , AUTO REGELNR. , EPROM PROG. , JOEDEL , HEXDUMP , FOURIER , ENKEL OF DUBBEL , MICRO 80 , ALLCOS , BATCHER , PRETPOOK , WORDPROC. , GEINTJE , TALSTELSELS , MIRAKEL.

Ik dank Wim Ernst hartelijk voor het tikwerk dat hij heeft gedaan voor dit bandje.

BAND 12 :

INDEX

HAMSAT TOEL.	: dit is een hulp programma bij het HAMSAT programma, hierin wordt het een en ander verklaard.
HAMSAT-83	: dit is een programma waarmee gegevens van diverse satelieten kunnen worden berekend.
MOONBOUNCE	: dit progamma stelt u in staat om de positie van de maan op een willekeurig tijdstip te bepalen.
AFST. BEREK 1	:
AFST. BEREK 2	: deze twee programma's berekende afstand tussen twee punten op aarde.
QTH LOCATOR 1	:
QTH LOCATOR 2	: deze twee programma's doen het zelfde als de twee voorgaande, maar dan uitgaande van de zendamateur indeling(coordinaten).
REF. OMLOOP	: dit programma doet in feite hetzelfde als moonbounce maar dan voor satelieten.

BAND 13 :

INDEX

Voor het samenstellen van deze twee bandjes ben ik de heren H v/d Heyden (PA0-AXA) en H Blijleven (PA0-HBL) zeer erkentelijk en ik hoop dat nog menig zend en andere amateurs er veel plezier aan zullen beleven.

Zelf ben ik bezig met het samenstellen van een utilitie bandje waarop onder andere de volgende programma's komen te staan:
Een naar mijn mening goede snelle database met een snelle sorteer routine, een eigen COS (fast en slow) en goede print mogelijkheden.
Dan de 40*24 soft-VDU die vorige keer ook al aan gekondigd is.
En nog enkele kleinere programma's.
Daarnaast komt er een bandje met leuke spelletjes voor de liefhebbers.Dit worden zowel behendigheids als denk spelen.
En in mijn achterhoofd ben ik ook al bezig met een grafische en wiskunde band.
Als je iets hebt waaarvan je denkt dat is wel de moeite waard, stuur het dan naar je regioarchievaris, die zorgt er dan wel voor dat het bij mij komt.
Als je denkt, dit programma heeft mij veel tijd gekost ik geef het liever niet zomaar weg, neem dan even kontakt met mij persoonlijk op, er is altijd wel wat te regelen.

Hierna volgt nog een lijst met de regionale archieven.

De ATOMaire bandenboer

N. Stad

ADRESSEN VAN DE REGIONALE ARCHIEVEN:

NOORD-NEDERLAND	:	S.Hoekstra A. Tasmanplein 50 9726 EM Groningen
OOST-NEDERLAND	:	F v/d Biezenbos Burcht 75 7608 JC Almelo
OVERIJSEL/GELDERLAND	:	H v/'t Riet Grevelingen 12 8032 KT Zwolle
UTRECHT	:	H Seymonsbergen F. de Hooghlaan 37 3741 RL Baarn
NOORD-HOLLAND	:	N Stad Plataanweg 47 1544 PB Zaandiijk
DEN HAAG	:	H v/d Heyden Pr. Annalaan 574 2263 XZ Leidschendam
DELFT	:	A Bliek Noordeinde 43 2611 KG Delft
ROTTERDAM	:	G Akkermans De Pintostraat 66 3052 NP Rotterdam
ZEELAND	:	nog niet bekend
ARNHEM E.O.	:	C Uges W. de Withstraat 78 6712 HE Ede
OOST-BRABANT	:	C Kwakernaak v. Speycklaan 29 5694 CG Breughel
VENLO E.O.	:	G Borghaerts Hatertseweg 3 6581 KD Malden
LIMBURG	:	J Smeets Resedastraat 28 6163 TP Geleen
BELGIE	:	J Mijngheer Fr. Verdonckstraat 39 B.8 1140 Brussel

Mochten er onjuistheden in deze lijst staan laat het dan weten, dan kan ik de volgende keer een bijgewerkte lijst publiceren. Als er in de toekomst veranderingen optreden geef het dan zo snel mogelijk door.
Dat was het dan voor deze keer.

N.Stad
075-280808

TURTLE GRAPHICS VOOR DE ACORN ATOM

In diverse hobbycomputertijdschriften ben ik verhandelingen tegengekomen over de zogenaamde TURTLE GRAPHICS. Geïnteresseerd geraakt door de aldaar genoemde mogelijkheden van deze bijzondere vorm van computer graphics heb ik getracht ook zoiets te maken voor de ACORN ATOM.
TURTLE GRAPHICS is feitelijk een onderdeel van de hogere programmeertaal LOGO, deze turtle graphics omvatten de grafische commando's van deze taal.
LOGO is evenals LISP en FORTH een zgn."threaded code language", d.w.z. men kan met een aantal basiswoorden nieuwe woorden definiëren, op welke wijze zeer krachtige mogelijkheden aan de taal worden toegevoegd.

Bij turtle graphics bestuurt men a.h.w. een cursor (de "turtle") met een aantal vrij eenvoudige woorden (commando's),alhoewel de basis simpel is, kunnen toch complexe grafische figuren gecreëerd worden.
In principe zijn er twee mogelijkheden om met turtle graphics te werken:

- direct (dus directe uitvoering van het ingevoerde commando)
- indirect, ofwel m.b.v. geprogrammeerde instructies

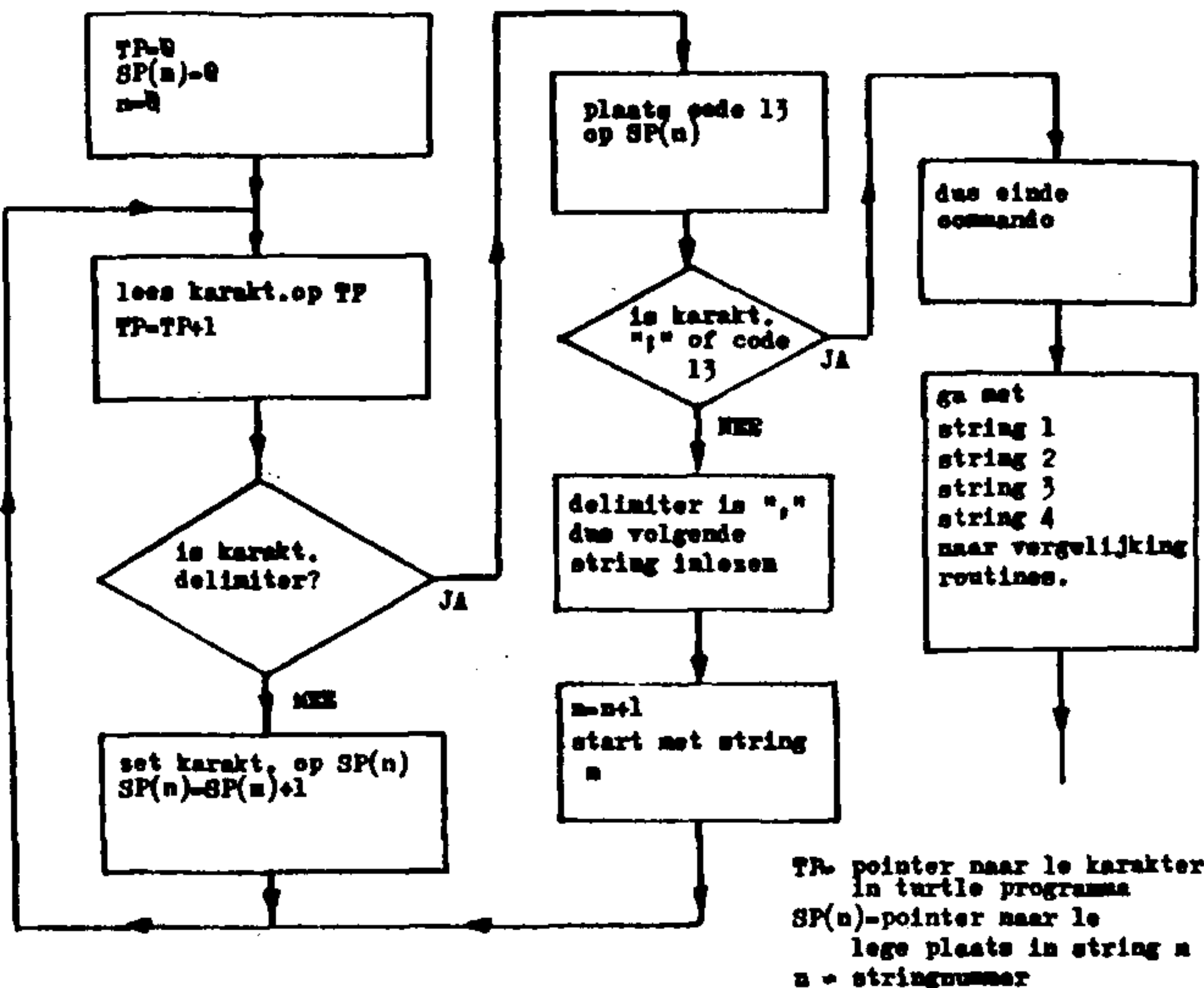
Voor deze laatste mogelijkheid heb ik in deze versie gekozen, vooral ook omdat de ATOM zich daar goed voor leent.

Om een dergelijke"taal" te laten functioneren heb je het volgende nodig:

a.de mogelijkheid om het turtle-programma in te voeren en te editen,
b.een vertaalprogramma, welke de turtelcommando's in het turtleprogramma omzet naar (voor basic) begrijpelijke strings, en
c.een aantal basic subroutines, welke de uiteindelijke uitvoering van de turtle-commando's bewerkstelligen.

ad.a Is goed mogelijk door het turtleprogramma in een ander tekstgeheugen te schrijven (dus gewoon voorzien van regelnummers), dit m.b.v. de tekstpointer (?18). Editen en listen is dan mogelijk zoals bij een normaal basic programma. De enige voorwaarde is dat je ervoor moet zorgdragen, dat de basic-interpreter bij een RUN niet probeert het turtleprogramma te interpreteren. We laten de interpreter daarom meteen springen naar de le regel van het eigenlijke basic (stuur-)programma.
Dit heb ik zelf geregeld door in de le regel van het turtleprogramma te schrijven: LINK Q, dit begrijpt de interpreter en springt naar de MCode-routine op adres Q, deze draagt zorg voor het feit dat de interpreter naar het eigenlijke basic-programma springt.
ad.b Dit vertaalprogramma schrijf je dus gewoon in basic in het normale werkgeheugen (meestal vanaf #2900), hoe dit vertaalprogramma werkt is in het blokschema aangegeven. Het programma produceert strings en test deze met een aantal referentie-strings, waarna een bijbehorende subroutine wordt aangeroepen.
ad.c de subroutines dragen zorg voor de uiteindelijke uitvoering van het commando, het turtle-commando gaat meestal vergezeld van een (aantal) parameter(s), welke eveneens in strings worden omgezet.
Deze parameter-strings worden dan veelal gebruikt voor het veranderen van een aantal variabelen, zoals bijvoorbeeld de cursorpositie ed.

De clou van deze manier van werken is dus dat je schijnbaar een turtleprogramma laat uitvoeren, maar onze ATOM eigenlijk niets anders laat doen dan een basic-programma runnen.



nu over de turtle-graphics zelf:

een turtle-commando heeft het volgende format:

COMMANDO(,PARAM 1(,PARAM 2(,PARAM 3)))

de parameters zijn tussen haakjes aangegeven, of zij wel of niet benodigd zijn hangt af van de aard van het commando, het maximum aantal parameters is 3. Parameters worden van het commando gescheiden door een komma, ook tussen de afzonderlijke parameters wordt een komma gebruikt, (als delimiter). Commando's afzonderlijk mogen in één regel, gescheiden door een puntkomma (als in atombasic),naast elkaar gezet worden. In dat geval werkt de puntkomma als delimiter tussen de commando's, ook een 'return' (code 13) werkt op die manier als delimiter.
In mijn versie mag je zowel de volledige commando's ingeven, als de afkorting van het commando (per definitie altijd de le 2 karakters van de volledige naam). Dit geeft een redelijke snelheidsverhoging. Tenslotte is het vertaalprogramma in basic geschreven zodat je eigenlijk niet van "snel" mag spreken.
De commando's welke ik heb gedefinieerd in het vertaalprogramma zijn de volgende:

Commando	omschrijving
GMODE,n	kies graphics mode n (n tussen 0 en 7, incl. colourmodes)
CHMODE,n	verander graphics mode naar n (zonder clear screen)
CLS	clear screen (in huidige graphics mode)
ON	zet cursor aan, d.w.z. tekenen bij beweging
OFF	zet cursor uit, d.w.z. alleen bewegingen
CCLR	zet cursor in clear mode, d.w.z. wist uit waar hij zich bevindt
CNV	zet cursor in inverteer mode, dus inverteren bij beweging
COL,n	kies colour n (cursor)
ANG,n	zet de richting van de cursor in hoek n (0-360 graden)
ROT,n	roteer de cursor n graden (antiklokagewijs)
FWD,n	verplaatst de cursor n punten in huidige richting
BWD,n	idem, in tegengestelde richting
RIGHT,n	beweeg cursor n punten naar rechts, ongeacht richting
LEFT,n	idem, naar links
UP,n	idem, omhoog
DOWN,n	idem, omlaag
CIRCLE,n,m,R	teken cirkel vanaf hoek n tot hoek m, straal R
VAR,X,n	definieer variable X als waarde n (X van A tot Z)
ADD,X,n	tel n op bij variable X
SUB,X,n	trek n af van variabele X
DIV,X,n	deel variabele X door n
MUL,X,n	vermenigvuldig variabele X met n
SQFILL,n,m	vul een blok op positie van cursor,lengte n, breedte m
SQCLR,n,m	clear een blok op positie van cursor, idem
SQINV,n,m	inverteer een blok op positie van cursor, idem
INV	inverteer het gehele scherm
MIRR	spiegel het gehele scherm
HOME	zet cursor op positie 0,0
CKEY	wacht tot toets is ingedrukt
WRITE,n,m,$i	schrijf $i op positie n,m
PLAY,n,m	laat toon horen, toonduur n, toonhoogte m
PAUSE,n	wacht gedurende n (n in 1/10 seconde)
END	einde turtleprogramma (verplicht)
MOVE,n,m	verplaatst cursor naar absolute positie n,m
REPEAT,n	herhaal commando's tot UNTIL, n keer
UNTIL	functioneert als end repeat
DEFINE,$i	definieer $i als de commando's tot aan CLSDEF
CLSDEF	functioneert als end define
CALL $i	roept de gedefinieerde $i aan (gedefinieerd woord)

n en m zijn integere waarden, maar mogen ook gedefinieerde variabelen zijn (dus A t/m Z), $i is altijd een string.
Vooral de commando's REPEAT/UNTIL en DEFINE/CLSDEF/CALL zijn zeer krachtig, zie de voorbeelden aan het einde van dit verhaal.

Om een idee te geven van een van de gebruikte subroutines, bekijken we de uitvoering van het commando ROT,n:

wanneer het vertaalprogramma heeft geconstateerd dat het gaat om een ROT commando (1e string bevat ROT, 2e string bevat n), voert het de volgende subroutine uit:

```
1000 REM subroutine ROTATE
1010 I= VAL $ SS1
1020 %H=%H+I: %X=COS%H; %Y=SIN%H
1030 R.
```

Hierin is: %H de hoek waarin de cursor wijst (in graden)
$ SS1 is string nummer 2 (bevat n)
I krijgt de value van $ SS1
%X is de X-richting van de cursor
%Y is de Y-richting van de cursor (beide tussen -1 en +1)

let wel: het eigenlijke programma werkt nog ietwat anders, namelijk vóórdat aan I de value van $ SS1 wordt toegekend, wordt gekeken of $ SS1 niet een variabele voorstelt.

Een voorbeeld hoe je met turtle graphics een hexagoon tekent in mode 4:

```
100 LINK Q
110 GMODE,4
120 OFF
130 MOVE,100,100
140 ON
150 REPEAT,6
160 FWD,30
170 ROT,60
180 UNTIL
190 END
```

Fraaier werkt het met een definitie:

```
100 LINK Q
110 GMODE,4
120 OFF
130 DEFINE,$POLYGONE
140 REPEAT,A
150 FWD,30
160 ROT,B
170 UNTIL
180 CLSDEF
200 MOVE,100,100
210 ON
220 VAR,A,6; VAR,B,60
230 CALL,$POLYGONE
240 VAR,A,10;VAR,B,36
250 CALL,$POLYGONE
260 END
```

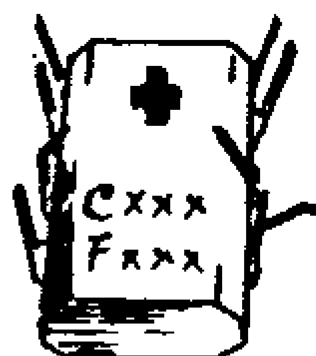
Dit turtle programma tekent op positie 100,100 een polygoon met 6 hoeken en een polygoon met 10 hoeken.

De mogelijkheden zijn legio, helaas kan ik de listing van mijn programma hier niet geven (neemt ca. 5,5 KByte in beslag).In mijn versie heb je dus 5,5 KByte normaal werkgeheugen nodig, vanzelfsprekend 6KByte graphics geheugen en dan nog geheugen voor het turtle programma (B.V. met een 16 K-kaart vanaf #4000 o.i.d., tenslotte moet nog wat geheugen (ca.250 bytes) beschikbaar zijn voor strings-opslag).
Mocht iemand geïnteresseerd zijn in het programma, laat hij dan contact opnemen met

R.J. Snelders
Welhaak 63
9932 BK Delfzijl, tel.05960-28367
Acorn Club Noord



Heden 1-9-1983 eindigde na een veelbewogen leven vol beproevingen, de produktie van ons aller vriend



ATOM geb. ACORN

Zijn gedachtenis zal in zegening blijven.

BBC	geb. Acorn London
FACTUM	geb. Acorn Malden
ELECTRON	geb. Acorn Cambridge

Gelegenheid tot condoleren op 26-27 november

te

Rendeux Haut, België

Namens U allen,

Hij ruste in vrede. Borghaerts

In Memoriam

Een goede vriend is van ons heengegaan.

Hoewel wij zijn einde hebben zien aankomen, treft ons bovenstaand bericht tóch smartelijk. En vragen wij ons af 'waarom zo vroeg ?' Hij was nog zo jong !

Hij was een goede vriend en een geboren leermeester. Met weemoed herdenken wij de eerste lessen, waarin hij ons het verschil tussen een bit en een byte probeerde uit te leggen. Echter gelukkig, van al zijn goede eigenschappen stak er één boven alles uit: zijn onuitputtelijk geduld. Bij ons wel hard nodig; velen weten het verschil nóg steeds niet.

Hij gaf ons veel en vroeg weinig; 5 Volt slechts en zelfs dát gaven wij hem niet altijd. Daarentegen was zijn vrijgevigheid allom bekend. Vroegen wij om één "R", dan gaf hij er met vreugde vele.

Inderdaad; een goede vriend is van ons heengegaan. Het verdriet grijpt ons aan. Maar richten wij ons spoedig weer op zoals de overledene ook gewild zou hebben. En zien wij om ons heen !

En wat zien wij dan ?
De waarheid, vrienden ! !

Die waarheid ís toch, dat velen het verscheiden van onze vriend niet hebben kunnen afwachten. En juist hém, de ATOM -wat onsplijtbaar betekent- tóch in stukken sneden en in onpersoonlijke Europrintjes in racks achter tralies zetten ! Die waarheid ís toch dat zijn jonger broertje Beeb niets nalaat om hem, Atom, in de ... (INPUT "zekere vloeistof"$A) te zetten. Een nóg jonger broertje 'Electron' probeert, zelfs ongeboren, reeds zijn plaats in te nemen. We moeten het nog zien !

En nóg een broertje,FACTUM, spoelde hem de hersens, negeerde zijn moedertaal en trok hem mee naar de MICRO-SOFT familie !

Laten we eerlijk zijn, beste vrienden; we lieten niet veel van hem heel ! En het werd ook tijd, dat onze goede vriend ons ging verlaten, nietwaar ? Hij benauwde velen van ons met z'n weinige Array's; met z'n zgn. 'afwijkende' stringbehandeling. Met z'n miezerige printsporen en met z'n stal vol gestapelde varkens, slurpende aan steeds grotere voedingen

Hebben wij zélf, bedroefde medeleden, hem niet tot bartstens toe opgeblazen door er steeds meer kaarten in te persen ??

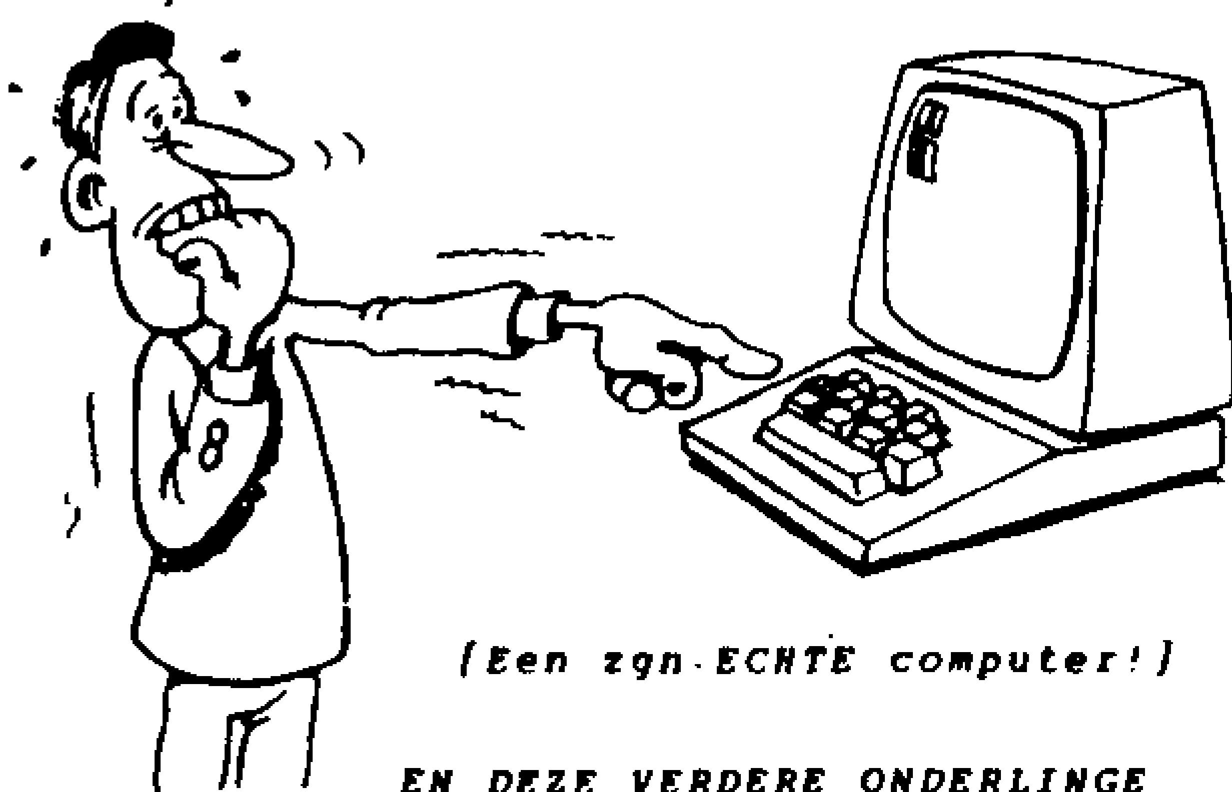
En nu we tóch eerlijk zijn, beste vrienden, koesteren velen van ons aan zijn graf, naast een diepe droefenis toch óók niet een heimelijk gevoel van vreugde ? Zo van 'Dat is dan dat en nu zijn we van hem af'? Zo is het toch, nietwaar ?

En aan zijn graf en letterlijk over zijn lijk bouwen dezulken een imitatie Apple, of kopen voor een krats een CBM-64, waar veel van hun dromen al uitgekookt inzitten. Inderdaad: zo van "En nu óp naar een ECHTE computer !

"Such is life" zou onze vriend zeggen, ware hij nog in leven. En met zijn fameuse geduld wacht hij af tot het ook tot deze hardlopers begint dóór te dringen, dat onze Atom eigenlijk toch wel een knappe en veel handiger computer was. En in meerdere opzichten z'n tijd nog steeds vooruit. En dan komt het heimwee. Naar een gebufferde bus met 'alles d'ran', naar een geheugenkaart met 'alles d'rop' En géén Monitor van Roel; géén Hex van Bram; géén Josbox etc op eenschakelkaart. En als maar: PRINT:PRINT etc in plaats van ''' enz enz

Deze hardlopers,beste vrienden, komen in zeer grote problemen. Na het verscheiden van de Atom zal hun aantal stellig toenemen.

ZIJ BEHOEVEN ONZE STEUN !



(Een zgn ECHTE computer!)

EN DEZE VERDERE ONDERLINGE STEUNVERLENING,BESTE VRIENDEN IS DE TAAK DIE ONS WACHT !!

Het gaat om onze TOEKOMST !

* * * * * * * * * * * * * * *

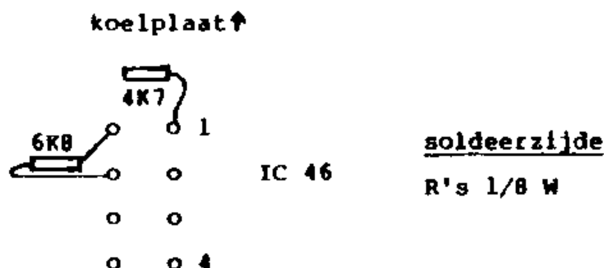
Doen wij niets, dan zal onze Vereniging na verloop van tijd uiteenvallen in net zoveel groepen, als er vervangende computers zijn. Willen we dat niet, dan zetten we de samenwerking voort; houden de 'weglopers' binnen de Club en proberen in andere computers, gebouwd of gekocht dátgene te realiseren, dat wij aan ATOM- eigenschappen hebben leren waarderen. Waarmee dus gezegd:

ATOM IS DOOD ; LEVE ATOM !

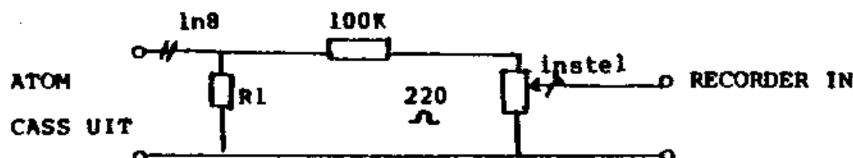
Borghaerts

6000 Baud recorderprogramma.

Naar aanleiding van de 3000 Baud routine is mij uit het Noorden gevraagd of het nog sneller kan. Welnu, het blijkt te kunnen. De timing is opnieuw uitgerekend; het bemonsteren (LDA#B002) is vervroegd om onnodige drop-outs te voorkomen en het printen van de bloknummers is netter. Helaas is de ingangsversterker van de Atom te traag (de eindtorren van de op-amp's van ic 46 worden sterk in verzadiging gestuurd wat ze erg traag maakt).
Dit euvel kan eenvoudig worden verholpen door van de uitgangen (p. 1 en 7) weerstanden naar de +5V (p. 8) te solderen (zie tek.).



Ik heb het programma uitgeprobeerd op een oude 4-sporenrecorder van een grote fabriek uit het Zuiden des lands en het werkt feilloos. Op twee goedkope cassetterecorders ging het ook proma nadat onderstaand schakelingetje voor opname was gebruikt.
De automatische opnameregeling van deze recorders geeft een te sterke uitsturing bij 6 KHz. Ideaal is om de automaat uit de recorder te slopen maar ook bruikbaar is om met de instelpotmeter van het schakelingetje het ingangssignaal zo zwak te maken dat de VU-meter niet verder dan 30% (-10 dB) komt. Voor handgeregelde recorders geldt uiteraard ook dat niet hoger dan 30% geregeld mag worden.
R1 is bij slechtere recorders nodig om de balans tussen hoge en lage tonen te verbeteren. Met wat experimenteren moet het lukken !



R1 : 3000 Baud : 39 K
R1 : 6000 Baud : 10 K

Hieronder staan de veranderingen om de 3000 Baud routine om te zetten naar 6000 Baud.

Regel:

5 : 694 wordt 734	310 : achteraan NOP
10 en 11 : 46 - 47	340 : 32 - 15
50 : 127 - 150	360 : 25 - 8
70 : 63 - 30	370 : achteraan NOP
80 : NOP - weg	730 : 29 - 13
100 : 31 - 14	740 : NOP - weg

110 : achteraan NOP
120 : 28 - 11
190 : 70 - 113
210 : 63 - 29
220 : achteraan NOP
250 : 32 - 15
300 : 24 - 7
780 : 29 - 11
790 : NOP - weg
800 : LL10 - LL47
801 : LL47 NOP;NOP;clc;BCCLL10
830 : 88 - 120
840 : ..LL20;LDA#B002;..
870 : ..LL27;LDA#B002;..
940 : ..LL20;LDA#B002;..
970 : ..LL27;LDX#B002;STX165;..
1000 : STA175 rest weg
1015 : 12 - 4
1040 : 26 - 11
1070 : ..LL40;LDA#B002;..
1090 : NOP;BNELL38;JSRLL27;LDA#B002
1100 : LDA - LDX , CMP - CPX
1240 : 28 - 14
1280 : 12 - 7
1300 : 28 - 14
1330 : 14 - 8
1420 : :LL28 STA165;LDX@13
1425 : NOP;NOP;NOP;NOP;NOP
1450 : 18 - 10 , 26 - 10
1510 : :LL33 LDA#B002;LDX@4
1620 : 15B - 168

Toevoegen:

171 LDA#EO;SEC;SBC@4;STA#EO

891 idem

regel 1630 vervalt.

Deze toevoeging geeft een nettere print voor de bloknummers en is ook bruikbaar voor 3000 Baud; dan ook veranderen r. 5: 694 - 707 en r. 1620 : 15B wordt 162.

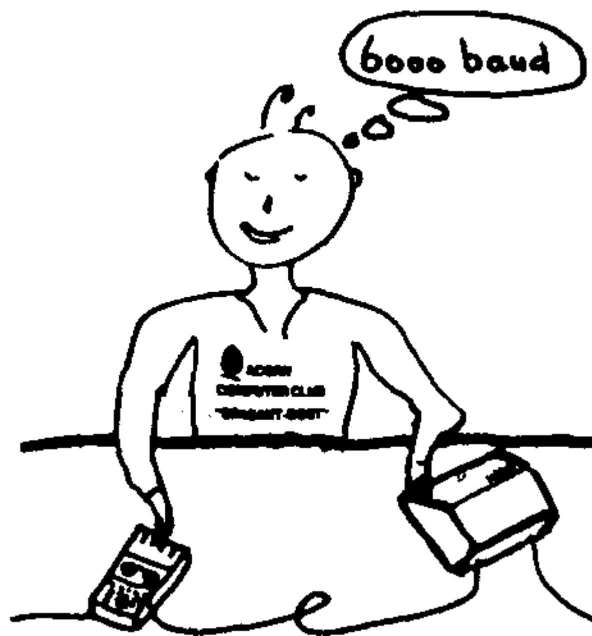
De routine kan overal geladen worden, echter, als een lus over een paginagrens ligt, is de timing niet meer optimaal.
Kontrole is mogelijk door de assembler te laten listen. Het is in ieder geval in orde als de routine beging op een paginagrens, b.v. #3900.

Overigens wil ik binnenkort een interpreter voor het AXXX adresgebied ontwikkelen met o.a. de 3000-6000 Baud routine die tevens operating system voor de schakelkaart is (ook zonder kaart te gebruiken). Dit systeem is flexibel en vereist geen verandering in F.P.ROM en hardware van de Atom. Het maakt geen gebruik van het EXXX adresgebied.

Peter Ehrlich

N.B. Voor 3000 Baud routine zie "de Cursor" no. 4 , jaargang 1

"Acornnieuws" no. 4, jaargang 2.



MJCOS --- EEN 9600 BAUD CASSETTE OPERATING SYSTEM

Met MJCOS heb je niet alleen een zeer snelle cassette interface, maar een uitgebreid Cassette Operating System, dat het standaard ACORN ATOM COS kan vervangen. De belangrijkste redenen om MJCOS te gebruiken zijn:

1. Snelle en kompakte registratie.

   De maximale snelheid is 9600 bits/sek. Dit is 32 maal zo snel als in de standaard COS. Kompakte registratie heeft tot gevolg dat de 16K RAM kaart in minder dan 20 sek. op tape staat! Dit is ongeveer 50 maal zo snel als in standaard COS.

2. Instelbare snelheid.

   Omdat de frequentie van het cassette signaal evenredig is met de bitsnelheid heb je voor 9600 bps helaas een goede recorder nodig. Met een eenvoudig kommando kun je echter de snelheid instellen op een lagere waarde: 4800, 2400 of 1200 bps. Voor elck wat wilsch dus. Voor 2400 bps bleek een zeer slechte recorder al goed genoeg.

3. MJCOS werkt feilloos.

   Op een PIONEER CT200 werden 1 miljoen (!) bytes weggeschreven op 9600 baud en vervolgens teruggelezen. Resultaat: geen enkel bitje verkeerd!

4. Cassette VERIFY.

   Als je een file van tape kunt lezen zonder SUM ERROR, kunnen er toch nog bits fout zijn, zeker bij de eenvoudige SUM check van de ATOM. Zekerheid krijg je alleen door de file op tape te vergelijken met de file in het geheugen. Dit gebeurt in MJCOS als je het VERIFY kommando uitvoert.

5. Betrouwbare controle op leesfouten.

   MJCOS kontroleert op een professionele manier op leesfouten. Voor de kenners: Cyclic Redundancy Check (CRC).

6. Meerdere files per LOAD/SAVE.

   Met een enkel SAVE kommando kun je meerdere files ineens naar tape schrijven, ook al staan ze niet achter elkaar in het geheugen. Voor het laden hiervan is een enkel load kommando voldoende.

7. Gebruik van filenamen.

   Het gebruik van filenamen heeft in MJCOS geen nadelige invloed op de snelheid. Bij het load kommando is een verkorte filenaam toegestaan, d.w.z. je hoeft alleen de beginletters van de gewenste file in te typen. MJCOS vergelijkt iedere letter van de filenaam die jij hebt ingetypt met de overeenkomstige letter van de naam die op tape staat. Als deze letters overeenstemmen, dan

[illegible] het laden. Geef je een 'dummy' filenaam (""), of helemaal geen naam op, dan begint het laden met de eerste de beste file. Een file die "TEKST23/08/83" heet kun je b.v. laden door "TEKST23" of "TE" op te geven.

8. TOP en ?18 automatisch gezet.

De problemen die te maken hebben met het feit dat de ATOM twee kommando's kent om een file te laden, behoren in MJCOS tot het verleden. Je hoeft je dus geen zorgen meer te maken over de text space pointers ?18 en TOP. Na het laden van een BASIC programma zorgt MJCOS er voor dat deze goed staan.

9. Verbeterde CAT.

Bij een CAT drukt MJCOS slechts EEN regel af per file (i.p.v. per blok van 256 bytes), met daarin alle gegevens over de file. Deze regel verschijnt bovendien meteen aan het begin van de file.

10. Visible load.

Uiteraard ontbreekt deze bij MJCOS niet. Bij het laden zie je in de linkerbovenhoek van het scherm alle bytes voorbijkomen.

Het bijzondere van MJCOS is dat er (afgezien van de VIA en een eenvoudig weergave filter) geen extra hardware nodig is om een 9600 bps cassette interface te krijgen. Alles draait op de software, die ondanks de vele extra's slechts (300)hex bytes lang is.

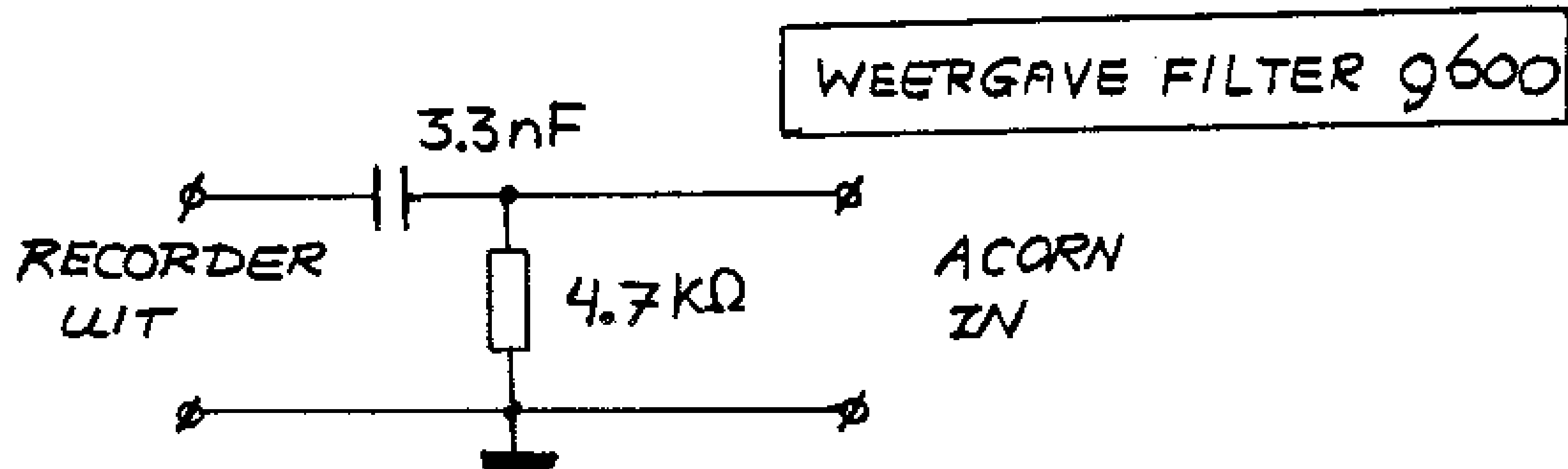
Hoe werkt het?

Het uitgangspunt bij het ontwerpen van MJCOS was om met een minimum aan wijziging van het bestaande systeem, een maximum aan funktionaliteit te realiseren. De enige 'inbraak' in de software is dan ook het veranderen van de innoud van de zgn. Command Line Interpreter vector, de lokaties #206 en 207 (COMVEC). Normaal staat in COMVEC het adres van de interpreter van de COS kommando's, zoals *SAVE, *LOAD etc.: #F8EF. Door het adres van de MJCOS interpreter in COMVEC te zetten, kun je alle kommando's die met een ster (*) beginnen doorsturen naar MJCOS. Dit gebeurt automatisch wanneer je 'linkt' naar het beginadres van MJCOS, b.v. LINK#7D00.
Als MJCOS het kommando niet herkent, dan stuurt hij dit alsnog door naar de originele '*-interpreter'. Hierdoor kunnen diegenen die COMVEC al gebruiken om de ATOM nieuwe kommando's te leren - zie ACORN NIEUWS 2.2 pag. 61 e.v. - dit nog steeds blijven doen. Ook kun je nog steeds via de cassette interface van de TOOLKIT, JOSBOX etc. blijven werken, zelfs wanneer MJCOS aktief is.

Wat heb je nodig?

1. De VIA.
2. Een weerstand van 4k7.
3. Eventueel een opname en/of weergave filter (afhankelijk van de recorder en van de snelheid die je wilt halen).

De weerstand van 4k7 komt tussen pin 17 van de VIA (PB7) en pin 1 van de DIN plug (SK2). Dit heeft geen nadelige gevolgen voor het gebruik van de VIA. In tegendeel, m.b.v. PB7 kun je komplexe signalen (geluidseffecten) produceren, en deze via het cassette snoer aansluiten op een versterker. De weerstand zorgt er voor dat PB7 nog steeds bruikbaar is voor andere doeleinden.
Voor de filters kan ik helaas geen algemeen recept geven, aangezien deze afhankelijk zullen zijn van de gebruikte recorder en van de bitsnelheid. Het onderstaand hoog(!)doorlaatfilter doet het op mijn cassette deck erg goed bij weergave op 9600 bps. Een opname filter werkt op deze snelheid averrechts.



De kommando's.

Deze zijn min of meer een uitbreiding van de bekende ATOM COS kommando's. Verder moet je MJCOS kommando's altijd afkorten tot EEN letter. Het save kommando is iets ingewikkelder/krachtiger geworden, waardoor load/run eenvoudiger is dan vroeger. Een load komt vele malen vaker voor dan een save, dus zo hoort het ook.

*S."naam"SSSS LLLL EEEE,"naam"SSSS LLLL, . . . ,"naam"SSSS LLLL

Dit is de meest uitgebreide vorm van het save kommando. Hiermee kun je een of meerdere 'files' op tape zetten. Een file kan een BASIC of assembler programma zijn, of gewoon data. Net als in standaard COS is:

naam	naam van de file, Ø - 13 karakters
SSSS	beginadres
LLLL	eindadres plus een
EEEE	executieadres

De basisvormen van het save kommando zijn:

1) *S. "naam"
2) *S. "naam" SSSS LLLL C2B2
3) *S. "naam" SSSS LLLL
4) *S. "naam" SSSS LLLL EEEE

Met 1) save je een 'normaal' BASIC programma. Dit kommando komt in de plaats van het oude SAVE (zonder *). Met 2) save je een BASIC programma dat aan het eind nog een stukje assembler code of data bevat. Een dergelijk programma moest je voorheen ook al saven met het *SAVE kommando. Door het intikken van C2B2 als executie adres vertel je MJCOS dat dit (evenals bij 1) een BASIC programma is. MJCOS heeft deze informatie nodig om tijdens het laden de BASIC pointers ?18 en TOP te kunnen zetten.
Met 3) en 4) save je een machinecode programma of gewoon data. Dit is gelijk aan het oude *SAVE.

Achter al deze basisvormen kun je nog een willekeurig aantal malen (,"naam" SSSS LLLL) intikken. Hierbij is SSSS het startadres en LLLL het eindadres van een extra file die je tegelijk met het hoofdprogramma wil saven (en later ook laden'), b.v.

*S. "CHESS" 2800 3C00 378A, "DATA" 9000 9800

Geef je geen naam op dan krijgt deze file dezelfde naam als z'n voorganger. SSSS en LLLL zijn uiteraard verplicht. Bij het saven van zo'n extra file verschijnt er geen 'RECORD TAPE' meer. Je laat de band gewoon doorlopen tot de BASIC prompt '>' weer verschijnt. Als dit het geval is staan alle files op tape.

Omdat het input buffer van de ATOM ongeveer twee regels lang is, kun je op deze manier ongeveer vijf files tegelijk saven. Wil je deze serie verder uitbreiden dan kan dat ook. Gewoon na het laatste eindadres een extra komma (,) intypen, b.v.

*S. "SPACE-WARS" 2900 3C00 35C7, "VDU RAM DATA" 8000 8600,

Als deze serie op tape staat stop je de recorder en gebruik je het *X. kommando i.p.v. *S. om de resterende files te saven, b.v.

*X. "UTILITIES" 6000 6800, "PAGE2 DATA" 0200 0400

Omdat het hier nog steeds extra files betreft zijn zowel SSSS als LLLL verplicht en EEEE verboden.

*L. "naam"

Hiermee kun je een BASIC programma, assembler programma of gewoon data laden. De naam van de file kun je afkorten of weglaten, zoals beschreven in punt 7 van de inleiding. Daarnaast kun je nog eventueel een 'relocation' adres opgeven. Standaard gaat de file naar de plaats waar ie stond bij het saven.
Na het laden van een BASIC programma (executie adres = C2B2) zet MJCOS de BASIC pointers ?18 en TOP op hun juiste waarde, ook als je een 'relocation' adres hebt gegeven. Zoals beloofd laadt MJCOS automatisch alle files die als extra file werden meegesaved. Deze files komen altijd op hun oorspronkelijk adres in het geheugen, en hebben verder geen invloed op ?18 en TOP.
Als MJCOS de eerste file uit een serie files mist, dan verschijnt de mededeling 'REWIND TAPE'. In dit geval de tape terugspoelen en een willekeurige toets indrukken om het laden te hervatten.

*R. "naam"

Met dit kommando kun je een machinecode programma laden en automatisch RUNnen. Het laden gaat op dezelde wijze als bij *L.
Een BASIC programma kun je niet automatisch runnen. *R."naam" en *L."naam" doen in dit geval precies hetzelfde.

*V. "naam"

Hiermee kontroleer je of een (serie) file(s) goed op tape staat. MJCOS vergelijkt ieder byte van de tapefile met het overeenkomstige byte in het geheugen. Is er ergens een verschil dan volgt onmiddellijk de mededeling 'SUM ERROR'. Dit betekent niet dat *V. alleen maar kijkt of de sumcheck in orde is!

*C.

Dit is de (gewijzigde) CAT funktie. Per file print MJCOS achtereenvolgens :
FILENAME (startadres) (eindadres) (executieadres) (file nummer).
Stoppen door CTRL toets in te drukken. Dit werkt echter alleen zolang er nog een cassette signaal is. Eventueel de tape een eindje terugspoelen of anders BREAK gebruiken. Na een BREAK moet je MJCOS opnieuw linken.

*0 hierna gaan alle *-kommando's naar originele interpreter.

*1 snelheid wordt 1200 bps
*2 idem 2400 bps.
*3 idem 4800 bps.
*4 idem 9600 bps.
Tevens schakel je hiermee het 2400 Hz signaal uit, dat de ATOM standaard op de DIN plug zet.

Slotopmerkingen.

1. MJCOS gebruikt van de zero page alleen de locaties die voor het COS zijn gereserveerd.

2. De BPUT, BGET, SPUT en SGET lopen nog steeds via de 'oude' COS.

3. Bij het assembleren van de source code voert het programma een dialoog met de gebruiker. Om te kontroleren of er geen typefouten zijn gemaakt, assembleert het programma eerst naar #9500, en berekent een 'checksum' van de gegenereerde code. Pas als deze goed is begint het eigenlijke assembleren. Na het beginadres vraagt het programma of de code naar ROM moet en zo ja, naar welk beginadres. De code wordt in het laatste geval automatisch aangepast (gereloceerd).

4. Met de gegeven source code staat MJCOS bij het opkomen altijd op 9600 bps. Dit wordt bepaald door de 'LDA@#34' instruktie in regel 330. Hier wordt de ASCII code voor het cijfer 4 geladen, hetgeen bij het opstarten een *4 simuleert. Wil je dat MJCOS opkomt in een andere snelheid, laat het assembleer programma eerst een keer proefdraaien, en verwijder regel 100 als alles goed is gegaan. Hierna kun je regel 330 veranderen en de definitieve RUN draaien.

5. Belangrijke routines in MJCOS:

label	rel. adres	funktie
GG0	0208	Lees aanlooptoon (high tone leader)
GG1	020E	Lees byte van tape (in A register en #C0)
TT0	0298	Zet aanlooptoon aan
PP0	02A2	schrijf byte in A naar tape

GG0, GG1 en PP0 saven alle registers.

Martin Janssen

```
10 REM MJCOS SOURCE CODE 21/08/83
20 REM (C) M.M.H.M. JANSSEN
30 REM       BUITENSTEDE 30
40 REM       3431 XB  NIEUWEGEIN
50 DIM A1,DD6,SS11,XX0,LL22,II1,FF4,GG20,TT0,PP13
60 FOR I=0 TO 85; DDI=#8000; N.
70 P.''"EVEN PROEF DRAAIEN..."$21
80 I=#9500;O=#1234;GOS.b;GOS.b;P.$6;O=0
90 FOR I=#9500 TO P-1;O=O+(I-#94FF)*?I;N.
100 IF O<>#26686D5 P.'"SOURCE CODE IS FOUT"; E.
110 P.''"assembleer mjcos"''"GEEF BEGINADRES..."; IN.I
120 IN.'"AANPASSEN VOOR ROM--J/N",$A
130 IF ?A<>CH"J" O=0
140 IF ?A =CH"J" IN.'"GEEF BEGINADRES IN ROM",O; O=O-I
150 P.'"WIL JE EEN LISTING?"''"1 = SCHERM"'"2 = PRINTER"
160 IN.'"(RET) = GEEN LISTING",$A
170 P.'"EVEN GEDULD A.U.B."'
180 P.$21; GOS.b; IF ?A=CH"2" P.$2$6
190 IF ?A=CH"1" P.$6
200 DOS.b; P?0=-68; P?1=-34; P?2=-18; P?3=-8
210 P?4=#20; P?5=#40; P?6=#80; P?7=0; @=3
220 FOR I=0 TO 7
230 P.'"        "&P+I,&P?I; N.; @=8;P=P+8; P.'$6$3
240 P.''"MJCOS STAAT IN HET GEHEUGEN"''"EIND ADRES "
250 P."VOOR *SAVE:"&P; END
260b P=I
270 B=#B002; C=#C0; D=#C4; E=#C7; F=#C8; G=#C9; H=#CB; J=#CC
280 K=#CE; L=#CF; M=#D1; N=#D3; O=#D5; R=#D8; S=#D9; T=#DC
290 U=#F876; V=#BE00; W=#F893; X=#EC; Y=#C3; Z=#206
300[
310:DD0 LDA Z+1;CMP @(DD3+O)/256;BEQ DD5;STA#FC;LDA Z;STA#FB
320 LDA @(DD3+O)&#FF;LDX @(DD3+O)/256::DD1 STA Z;STX Z+1
330 LDA @#34;BNE DD4::DD2 LDA#FB;LDX#FC;BNE DD1
340:DD3 LDY @0;STY B;JSR U;INY;CMP@#30;BEQ DD2;BCC DD6
350 CMP@#35;BCS DD6::DD4 TAX;LDA PP13+O-49,X;STA E
360 LDA PP13+O-45,X;STA F::DD5 RTS::DD6 LDX#100,Y;INY;STY N
370 CPX@#2E;BNE XX0;CMP@#58;BEQ SS0;CMP@#53;BNE LL0;CLC
380\
390:SS0 ROR H;JSR TT0+O;LDA#EA;BNE SS1;LDA@6;JSR#FC45
400:SS1 LDY N::SS2 BIT H;BPL SS3;JSR U;CMP@#22;BNE SS4
410:SS3 JSR#F818::SS4 LDX@L;JSR W;LDX@M;JSR W;BNE SS5
420 BIT H;BMI SS7+1;LDA#12;STA L+1;LDA#D;STA M;LDA#E;STA M+[illegible]
430 LDA@#B2;LDX@#C2;BNE SS6::SS5 BIT H;BMI SS6;LDX@J
440 JSR W;BNE SS7;LDA L;LDX [illegible]::SS6 STA J;STX J+1
450:SS7 LDA@0;STA H::SS8 JSR U;CMP@#2C;BEQ SS9;DEY;CLC
460:SS9 ROR H;INY;JSR FF0+O[illegible];ROR H;JSR U;CMP@#D;BNE SS2
470:SS10 LDA@D;STA V+11::SS11 RTS::XX0 JMP(#FB)
480\
490:LL0 CMP@#43;BEQ LL2;CMP@#4C;BEQ LL2;CMP@#5A;BEQ [illegible]
500 CMP@#52;BNE XX0;LDA@0::LL2 LSR A;[illegible] A
510 ROR A;STA R;JSR#F[illegible];LDX@J;JSR W;LDA#EA;BNE LL4;LDA@4
520:LL3 JSR#FC45::LL4 SEC::LL5 ROR S
```

```
530:LL6 JSR GG0+O;BCC SS11;LDX@0;STX T;:LL7 JSR GG1+O
540 STA#ED,X;CPX@13;BEQ LL8;INX;:LL8 CMP@#D;BNE LL7;LDX@8
550:LL9 JSR GG1+O;STA Q+1,X;DEX;BNE LL9;JSR GG1+O;BIT R
560 BVS LL11;LDA T;BNE LL13;BIT S;BPL LL16;LDY@#FF
570:LL10 INY;LDA(G),Y;CMP@#D;BEQ LL14;CMP#ED,Y;BEQ LL10
580:LL11 LDY@0;JSR#F992;LDX@L;JSR#F7EE;LDX@J;JSR#F7F1;LDA K
590 JSR#F802;JSR#FFED;:LL12 CLC;BCC LL6;:LL13 JMP#F9FD
600:LL14 BIT H;BVC LL15;LDA@5;BNE LL3;:LL15 LDA Q+2;BNE LL17
610:LL16 LDA L;STA Q;LDA L+1;STA Q+1;:LL17 BIT S;BPL LL18;CLC
620 ROR S;LDA J+1;CMP@#C2;BNE LL18
630 LDA L+1;STA#12;:LL18 JSR II0+O;:LL19 INY;JSR GG1+O
640 BIT R;BMI LL20;STA(Q),Y;BPL LL21;:LL20 CMP(Q),Y;BNE LL13
650:LL21 CPY#D4;BNE LL19;JSR GG1+O;LDA T;BNE LL13;INC L+1
660 INC Q+1;TXA;BNE LL18;LDA H;BMI LL12;LDA J+1;CMP@#C2
670 BNE LL22;JMP#CD9B;:LL22 LDA R;BNE II1;JMP(J)
680\
690:II0 LDY@0;STY T;DEY;CLC;LDA M;SBC L;STA#D4;LDA M+1
700 SBC L+1;TAX;BEQ II1;STY#D4;:II1 RTS
710\
720:FF0 STY N;JSR#FBB1;LDY@0;STY T;:FF1 LDA(G),Y;JSRPP0+O;INY
730 CMP@#D;BNE FF1;LDX@8;:FF2 LDA Q+1,X;JSR PP0+O;DEX;BNE FF2
740 LDA T;JSR PP0+O;:FF3 JSR#FE71;JSR II0+O;:FF4 INY;LDA(L),Y
750 JSR PP0+O;CPY#D4;BNE FF4;LDA T;JSR PP0+O;INC L+1;TXA
760 BNE FF3;INC K;LDY N;RTS
770\
780:GG0 STY Y;LDY@#80;BNE GG2
790:GG1 STY Y;LDY@#FF;:GG2 STX X;LDX@0;SEC;LDA@#20;AND B;STAD
800 BEQ GG1;:BNE GG13;:GG3 CPX E;BCS GG4;LDY@#80;BIT#B001
810 BVC GG20;:GG4 CPY@#FF;SEC;BNE GG8;BEQ GG19;:GG5 CPX E
820 BCC GG6;LDY@#FF;:GG6 BCC GG7;:GG7 NOP;:GG8 LDX@#FC;EOR D
830 STA D;:GG9 BNE GG10;:GG10 BNE GG13;:GG11 LDA@#20
840:GG12 DEX;BIT B;BEQ GG12;BCS GG14;CPX E;BCC GG15
850:GG13 DEX;BIT B;BNE GG13;BCS GG14;CPX E;BCC GG15;BCS GG12
860:GG14 INY;BMI GG3;BEQ GG5;DEY;BCS GG16;:GG15 EOR D;STA D
870:GG16 ROR C;LDA C;ROL C;ROL A;AND@1;EOR T;LSR A;BCC GG17
880 EOR@#AE;:GG17 ADC GG18;:GG18 STA T;NOP;LDX@#F5;INY;CPY@6
890 LDA D;BCC GG9;:GG19 LDX X;LDY Y;LDA C;STA#B000;RTS
900:GG20 CLC;RTS
910\
920:TT0 LDA@#C0;STA V+11;LDA@0;STA V+5
930\
940:PP0 PHA;STA C;STX X;STY Y;LDX@#FF;CLC;LDA V+4
950:PP1 LDA@#40 :PP  BIT V+13;BEQ PP2;STA V+13;BCC PF4
960 CPX@9;BCC PP2;LDX X;LDY Y;PLA;RTS
970\----------------------------------
980:PP4 INX;BEQ PP5;SEC;ROR C;:PP5 PHP;LDA#52;BCS PP6;ASL A
990:PP6 STA V+6;LDY @;SEC PHP;STY V+7
1000:PP7 ASL V+6;ROL V+7;BCC PP7;:PP8 CPX@#;BCS PP10;LDA C
1010 AND@1 EOR T;LSR A;BCC PP9;EOR@#
1020:PP9 STA T;:PP10 PLP;JMP PP1+O
1030:PP11\ 8 BYTES OF DATA
1040]
1050 RETURN
```

ANALOOG -- DIGITAAL CONVERSI
==============================

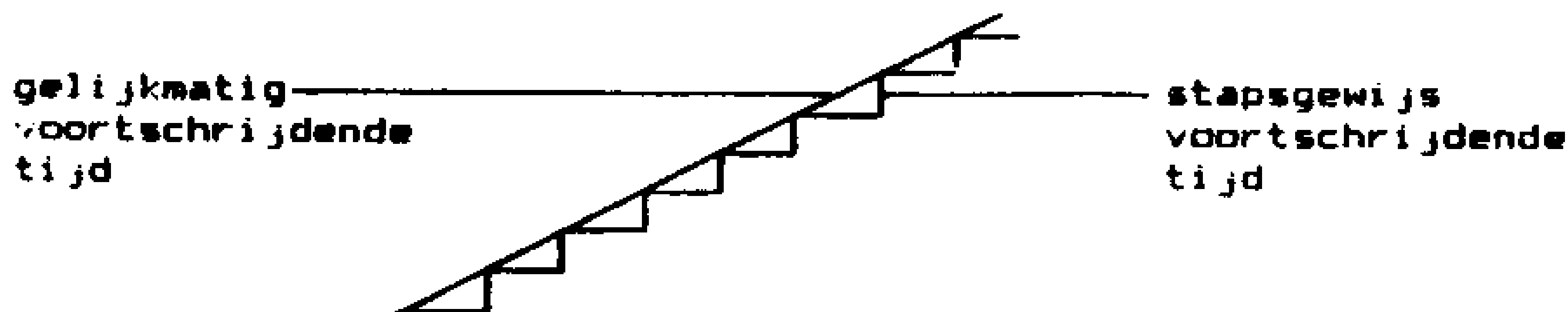
Het bovenstaande is voor niet-ingewijden een soort abracadabra; een toverformule die op zich onbegrijpelijk is maar die je soms nodig hebt en dan heel handige dingen doet.
Teneinde ook voor deze lezers (vermoedelijk de meerderheid!) de bijgaande twee artikelen althans in grote lijnen begrijpelijk te maken wil ik graag een tipje van deze sluier proberen op te lichten.

Allereerst enkele definities.
Een analoog (elektrisch) signaal is een signaal waarvan de spanning op wezenlijk dezelfde manier verloopt als de grootheid die wordt voorgesteld door dit signaal.
Een voorbeeld: Van een muziekinstrument, bijvoorbeeld een trompet, wordt een grammofoonopname gemaakt. Bij het afdraaien van deze grammofoonplaat beweegt de luidspreker en ook de pick-up-naald op dezelfde wijze als de lucht in de mond van de trompet, tijdens de opname. Daardoor beweegt bij de weergave van de grammofoonplaat de lucht bij ons oor ook op dezelfde manier, en daardoor horen we die trompet ook.
We kunnen het bovenstaande ook in grafiekjes weergeven:



Het zal u niet verwonderen dat de elektrische signalen in deze weergave-keten, zoals de uitgangsspanning van het pick-up element en de spanning op de luidsprekerklemmen er na passende vergroting c.q. verkleining precies zo uitziet als hierboven geschetst.

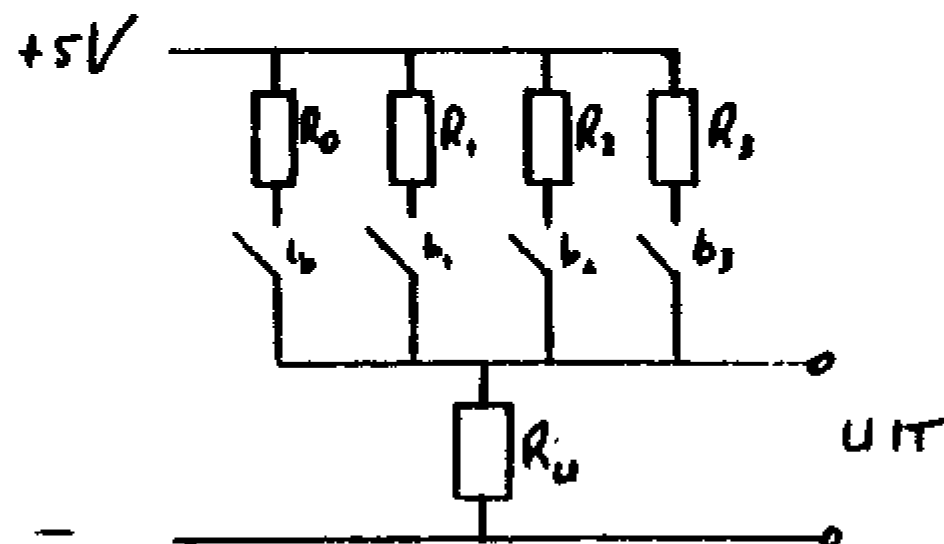
Een digitaal signaal bestaat uit een rij van getallen die een reeks opeenvolgende waarden voorstellen van de aangegeven grootheid.
Bijvoorbeeld een digitaal horloge vertoont een opeenvolgende reeks van momentopnamen van de tijd; deze schrijdt namelijk helemaal niet schoksgewijs voort (net zo min als de draaiende beweging van de aarde om haar as schoksgewijs verloopt) maar zeer gelijkmatig.
Zetten we de waarden van deze opeenvolgende getallenrij in een grafiek, dan ontstaat een aardige benadering van de voorgestelde grootheid:



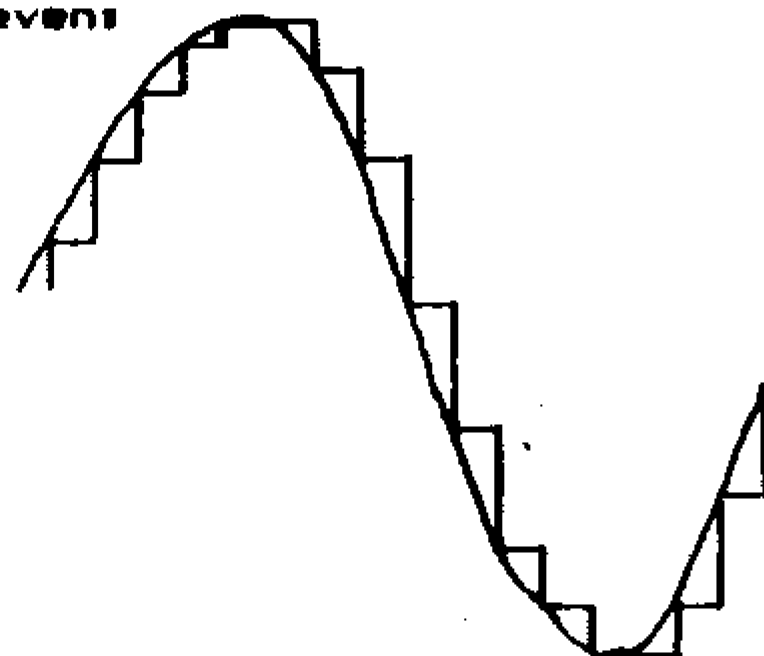
De precisie van deze benadering is afhankelijk van de stapgrootte. ofwel de grootte van het interval tussen twee opeenvolgende

getalswaarden. Is deze stapgrootte erg klein (bij een digitaal horloge bijv. tienden of zelfs honderdsten van een seconde) dan kan de benadering zeer precies zijn. Is deze stapgrootte erg grof (bijv. minuten) dan is de benadeing vrij slecht. De fout kan dan oplopen tot bijna 1 minuut, nl. 59,99999--- sec.

Het omzetten van een digitaal signaal naar een analoog signaal, dat dezelfde grootheid voorstelt, gaat op de manier zoals hierboven geschetst.
Dus de rij getallen wordt omgezet in een overeenkomstige reeks waarden (van bijv. een elektrische spanning of stroom). Aldus ontstaat een trapjeskromme, die een benadering vormt voor de voor te stellen grootheid. Zijn de trapjes erg klein, zoals bijvoorbeed bij geluidsweergave door middel van een compact disc, dan is het resultaat nauwelijks nog van het origineel te onderscheiden.
Zo'n digitaal-naar-analoog converter is in feite een erg simpele schakeling; gaan we uit van een binaire vorm van een getallenreeks met elk getal vier bits lang, dan is de volgende D-A-omzetter mogelijk:



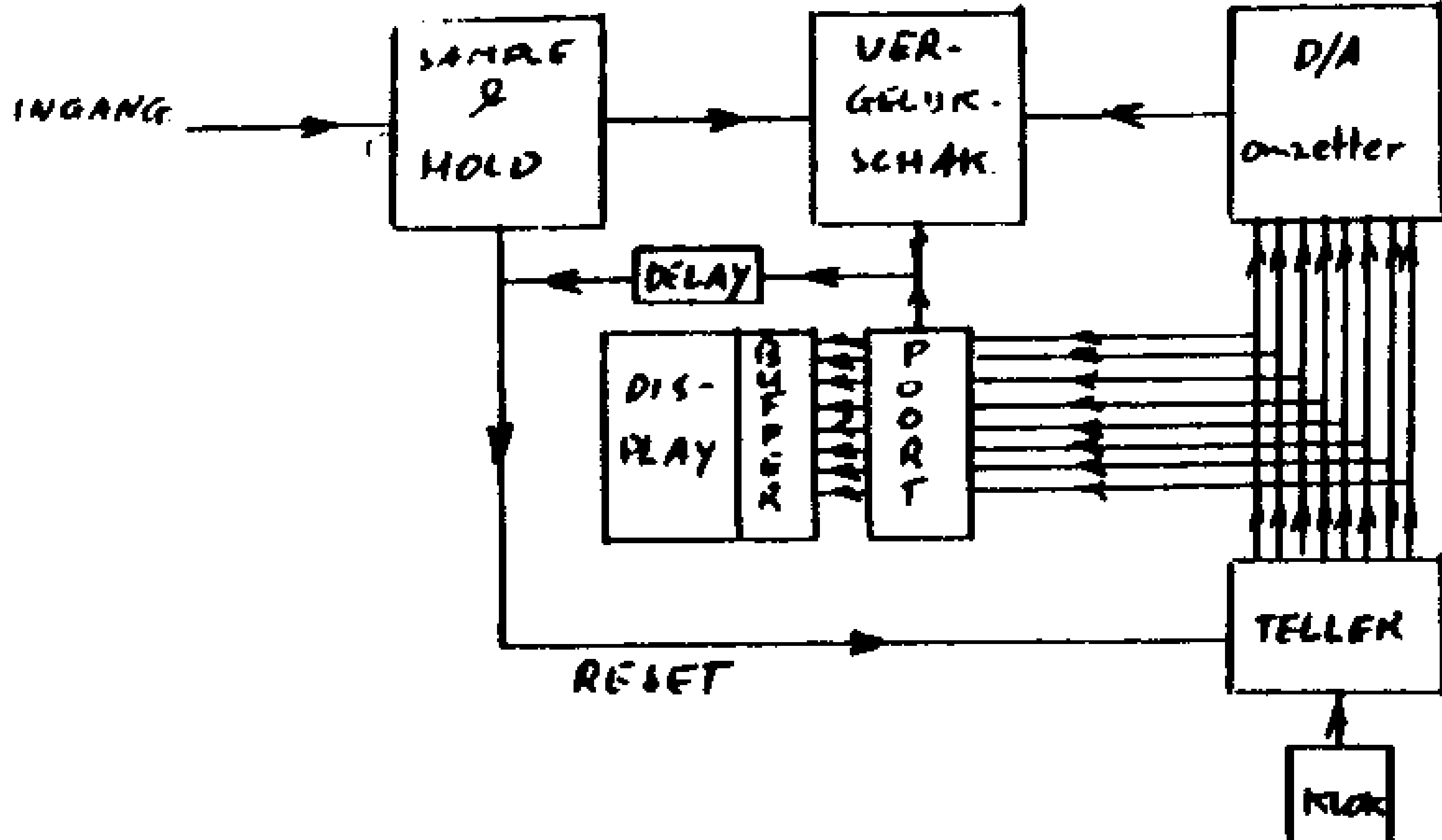
De schakelaars b0 t/m b3 worden bestuurd door het om te zetten binaire getal; b0 is het minst significante bit (waarde: 1) en b3 is het meest significante bit (waarde: 8). De weerstanden R0 t/m R3 zijn zodanig gekozen dat bij gesloten schakelaars R0 een stroom doorlaat met de waarde 1, R1 een stroom met waarde 2, R2 een stroom met de waarde 4 en R3 een stroom met de waarde 8 (U ziet, de schakeling kan eenvoudig worden uitgebreid met R4 en b4, R5 en b5, etc.). Al deze stromen vloeien samen door Ru, en veroorzaken daar een spanning, die evenredig is met deze totaalstroom. Dus bij alle schakelaars gesloten telt de uitgangsspanning voor 15, bij alleen b2 geloten voor 4, alleen b3 gesloten voor 8, enz. Bij alle schakelaars open heeft de uitgangsspanning uiteraard de waarde 0.
Op deze wijze zijn we in staat om bijvoorbeeld een sinusvorm met enige benadering weer te geven:



Dat een dergelijke benadering nog duidelijk "zanderig" klinkt, kunt u t.z.t. zelf konstateren aan de hand van het artikel "Toonkunst" (wat in de volgende Acorn Nieuws zal verschijnen), waar een 4-bits D-A-omzetter wordt gevormd uit vier "gewogen" VIA-uitgangen.

Een analoog naar digitaal omzetter gaat wat ingewikkelder te werk. Dat komt omdat eerst een trapjeskromme moet worden gevormd uit het "glad" verlopende analoge signaal. We moeten als het ware eerst "moment-opnamen" maken. Vervolgens moet elk van deze momentopnamen van het analoge signaal worden voorgesteld door de dichtstbijzijnde getalswaarde. Dit kan niet zomaar een willekeurig getal zijn, nee, deze getallen moeten gekozen worden uit de beperkte voorraad die kan worden voorgesteld met het beschikbare aantal cijfers of bits.
Dit wil ik even verduidelijken:
Stel dat we acht bits ter beschikking hebben: hiermee kan een getallenreeks worden gevormd van 0 t/m 255. Als de te digitaliseren spanning zich beweegt tussen 0 en 255 volt, dan ligt hiermee een klasse-indeling in een geheel aantal volt's vast. Elke tussenliggende waarde van een momentopname wordt dus benaderd door de direct hieronder liggende gehele waarde. Zo wordt een momentopname van 138,7 Volt voorgesteld door het getal 138. Een gering verlies van nauwkeurigheid dus.
We hebben nu dus de volgende twee behandelingen nodig voor een analoog-naar-digitaal omzetting:
1. Het maken van momentopnames; dit gebeurd in een zg. "sample-and-hold"-schakeling.
2. Het indelen van elke momentopname ("sample") in de beschikbare klassenreeks.

Dit laatste kan op allerlei verschillende manieren gebeuren: verreweg de eenvoudigste is die van het proberen: begin met de kleinste klasse; niet goed? probeer dan een klasse groter, enz. Totdat de goede klasse gevonden is. Dit "proberen" doen we nu met een teller, die evenveel bits breed is als de gewenste digitale precisie vereist.
Deze teller telt impulsen van een klok-oscillator, waardoor de tellerstand alsmaar opnieuw begint. De stand van deze teller wordt via een D-A-converter omgezet naar een elektrische spanning, die dus zaagtandvormig verloopt. Het is deze zaagtandspanning die vergeleken wordt met de analoge ingangsspanning. Zijn deze twee spanningen ongeveer gelijk aan elkaar, dan geeft de vergelijkingsschakeling een puls af naar de teller, die dan onmiddelijk stopt, met de gevonden waarde als tellerstand. Deze stand kan dan in een buffer worden overgenomen, waarna een nieuwe momentopname kan worden genomen en de teller weer kan gaan lopen en het hele spelletje weer van voren af begint.
Op deze wijze ontstaat dan de volgende schakeling (in blokschema):



De aldus gevormde A-D-omzetter is vrij traag: voor een omzetting moeten (gemiddeld de helft van) alle mogelijke klassen worden afgelopen. Bij een 8-bit brede omzetting is dit dus 128 klokpulsen. Met slimmere schakelingen kan dit beperkt worden tot precies evenveel klokpulsen als de omzeting breed is.
Overigens zijn er nog diverse andere typen A-D- en D-A-omzetters.

In de beide hiernavolgende artikelen wordt wat ''eper ingegaan op enkele practische A-D-conversieschakelingen en hun toepassing. Hopelijk hebben nu ook de niet-elektronisch geschoolde Atomisten enig idee van waar het over gaat.

Ton Otten.

Op deze lege plek even het volgende:
Uw redacteur heeft het wat drukker gekregen en heeft derhalve niet zoveel tijd meer om redacteur te zijn. Dit is dan ook een oproep aan hen die er voor voelen deze taak over te nemen. Mogelijk meerdere personen die het gezamelijk gaan doen.
U kunt natuurlijk rekenen op een ruime inwerkperiode in deze toch bijzonder mooie functie, waarbij je de inzage krijgt van alle hard- en softwarezaken van alle regio's.
Is er een gegadigde?
Gaarne bericht.
Dank u.

DAC / ADC

Hans Kept

Wie zijn computer wil koppelen aan de analoge omgeving kan nauwelijks om een Digitaal naar Analoog Converter (DAC) en een Analoog naar Digitaal Converter (ADC) heen.

Voor nog geen ƒ15.= koop je al een DAC0800 en hebt daarmee een i.h.a. ruim voldoend nauwkeurige converter (niet-lineariteit beter dan 0.2% volle schaal).
Rond deze DAC0800 heb ik onderstaande schakeling ontworpen en gebouwd.

IC1 is gebruikt als referentie terwijl IC3 fungeert als verschilstroom naar spanning omzetter.
Het geheel wordt aangesloten op poort A van de V.I.A.
De "offset" wordt weggeregeld door P2 zó in te stellen dat wanneer op poort A #7F resp. #80 wordt gezet gelijke, maar precies tegengestelde spanningen op de uitgang komen te staan (ca. - en +20 mV).
De "gain" wordt afgeregeld door P1 in te stellen op 5100 mV uitgangsspanning wanneer op poort A #FF wordt gezet.
Wanneer dit laatste niet lukt kan het nodig zijn R1 iets te wijzigen.
De stapgrootte van deze DAC bedraagt dan 40 mV (bereik -5100 mV tot +5100 mV), maar deze kan worden veranderd door wijzigen van R5 en R6.

Met zo'n DAC zou je bijvoorbeeld van je Atom een funktiegenerator kunnen maken voor sinussen, blokken, driehoeken, zaagtanden en wat er al niet meer aan spanningsvormen mogelijk is.

Veel groter worden de mogelijkheden als ook een ADC wordt gebruikt. Voor minder dan ƒ25.= kocht ik een ADC0804. Deze is redelijk nauwkeurig (beter dan +/- 1 LSB) en matig snel (in mijn schakeling ca. 10000 conversies/sec). Die snelheid kan hoger met andere typen ADC's, maar dan zul je waarschijnlijk heel wat dieper in je portemonnee moeten tasten en die 100 µsec/conversie kan je Atom nuttig gebruiken om ook nog iets met de binnengehaalde bytes te dóén.

Allereerst wordt het analoge ingangssignaal toegevoerd aan een verschilversterker (of liever: verzwakker) rond IC4.
Wil je een asymmetrische ingang, dan verbind je de min-ingang natuurlijk gewoon met de nul.
Vervolgens wordt het signaal bemonsterd door een "track&hold" schakeling rond IC5 en IC6. Condensator C11 wordt gebufferd door IC7 terwijl IC8 de feitelijke conversie voor zijn rekening neemt.
Deze schakeling wordt aangesloten op poort B van de V.I.A.
Op een opgaande flank op CB2 wordt de conversie gestart, terwijl de schakeling het einde van een conversie aangeeft door een negatief pulsje op CB1. Mijn V.I.A. gebruikt hiervan de opgaande flank om de byte in poort B te latchen.
De conversie-snelheid kan worden veranderd door C12 te wijzigen. De snelheid verder verhogen ging bij mij echter ten koste van de nauwkeurigheid.

fig.1.
Digitaal naar analoog coverter

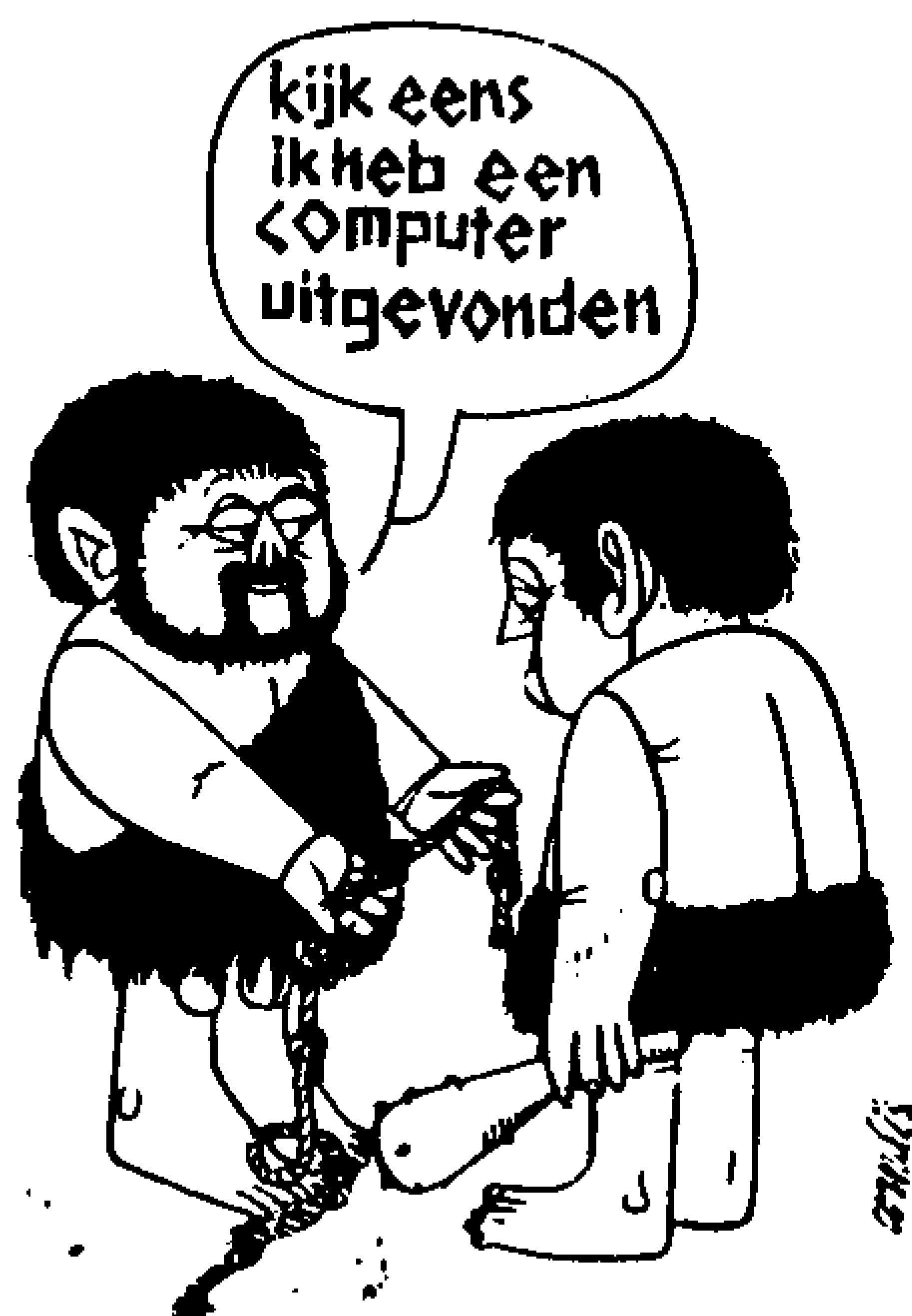
fig.2.
Analoog naar digitaal converter

De "offset" wordt zodanig weggeregeld, dat bij beide ingangen doorverbonden met nul, poort 9 schommelt tussen ∅7F en ∅80. De "gain" wordt zo afgeregeld dat 5060 mV ingangspanning ∅FE geeft op poort 8, d.w.z. ook weer 40 mV/stap. De stapgrootte kan veranderd worden door R9, R10 en R13 te wijzigen.

Door zijn conversie-snelheid van ca. 10000 conversies/sec is deze ADC theoretisch geschikt voor signalen met frekwentiekomponenten tot 5 kHz (denk bijv. aan middengolf zenders). Eventueel een filtertje voorschakelen.

Met deze converters kun je van alles doen. Om enige mogelijkheden te noemen:
- echo, phasing en vibrato produceren,
- een eenvoudig digitaal filter maken,
- spektraal-analyses (FFT) van allerlei signalen verrichten,
- de DAC en ADC laten samenwerken om zo diode- en transistorkarakteristieken op het beeldscherm te zetten,
- van je Atom een analoge oscilloscope maken, bijv. met de mogelijkheid om een plaatje naar het geheugen te kopieëren,
- en zo zijn er nog legio andere toepassingen te bedenken.

Clubgenoten die met deze schakeling willen experimenteren wil ik graag met raad en daad helpen.



Voorbeeld van een programma voor de DAC en ADC:

```
 10 REM ECHO
 20 B=#80;E#88:REM GEBRUIKT GEHEUGEN VAN BLOK B TOT BLOK E
 30 ?#B803=#FF;REM POORT A IS UITGANG VOOR DAC
 40 ?#B80C=#80;REM CB2 NEG. PULS BIJ WRITE, CB1 AKTIEF BIJ
 50            REM OPGAANDE FLANK
 60 DIM LL2,P(-1)
 70[
 80:LL0 LDX@0
 90 STA#B800 \START CONVERSIE
100 LDA@0;STA#A0;LDA@8;STA#A1 \INITIEER GEHEUGENWIJZER
110:LL1 LDA#B800;AND@#10;BEQLL1 \CONVERSIE KLAAR?
120 STA#B800 \START NIEUWE CONVERSIE
130 LDA#B800;CLC;ADC(#A0,X);RORA;STA#B80F;STA(#A0,X)\ECHO
140 INC#A0;BNELL1                \GEHEUGEN-
150 INC#A1;LDA#A1;CMP@E;BNELL1   \WIJZER
160 LDA@8;STA#A1;JMPLL1          \BIJWERKEN
170]
180 CLEAR 4;REM AKTIE ZICHTBAAR OP SCHERM
190 LINK LL0
```

Komponenten-lijst:

R1	56k2 met.film 1%	C1	1n	P1	100 (evt. 10 sl.)
R2	24k3 " " "	C2	47n	P2	22k
R3	2k15 " " "	C3	100n	P3	250 (" ")
R4	2k2	C4	10u/16V tant.	P4	10k
R5	2k37 " " "	C5	10n		
R6	2k37 " " "	C6	10u/16V	IC1	723
R7	100k " " "	C7	10u/16V	IC2	DAC0800
R8	100k " " "	C8	100n		(=LMDAC08)
R9	47k5 " " "	C9	100n	IC3	LF355
R10	47k5 " " "	C10	100n	IC4	741
R11	1k	C11	330p	IC5	4051B
R12	4k75 " " "	C12	39p	IC6	4093B
R13	4k75 " " "	C13	47n	IC7	CA3140
R14	4k75 " " "	C14	100n	IC8	ADC0804
R15	4k75 " " "				
R16	10k	D1	zener 6,8V		

COMPUTER VOLTMETER

(Het koppelelen van een 3-digit digitale voltmeter aan de acorn atom)

Het besturen van meetopstellingen met behulp van de computer is tegenwoordig in elk laboratorium een dood normale zaak. De apparatuur, die daarvoor gebruikt wordt is vaak onbetaalbaar. Is dit zonder al te veel kosten te realiseren op onze ACORN ATOM ?. Het antwoord op deze vraag is JA.
Wat hebben we nodig om dit te realiseren?
ten eerste: een Analoog-Digitaal omzetter, die al in vele electromica bladen gepubliceerd is. Ik heb de door Elektuur in Jan. 1979 gepubliceerde universele digitale meter gebruikt.
Ten tweede! Een interface, om de informatie, die uit de DVM komt in de computer te brengen. Deze interface zit reeds in de computer, nl. de 6522 VIA.
Ten derde: Een stukje software , dat de communicatie tussen de DVM, en de computer in goede banen leidt.

DE HARDWARE:
De DVM werkt als volgt! de ingangsspanning(-99....+999 mV) wordt in de CA 3162 mbv. "dual slope intregration" omgezet in drie 4-bits woorden, die de waarde van de 3 digits in BCD code voorstellen. Deze drie 4-bits woorden worden een voor een op de uitgangs (data) lijnen gezet. Tegelijkertijd wordt een van de 'digit select' lijnen hoog. De digit select lijnen geven aan voor welk digit de data bestemd is.(vergl. 'chip select lijnen' in onze computer). De displays lichten dus NA ELKAAR op. Dit 'multiplexen' is niet waar te nemen door de hoge frequentie (ca. 380 Hz).

HET KOPPELEN
Wat we nodig hebben voor het binnenhalen van de waarde die van de DVM af komt is het volgende: de waarde van ieder digit, het 4-bits woord dus, en een signaal, dt aangeeft voor welk digit de waarde , die door het 4-bits woord aangegeven wordt bestemd is. De digit select uitgangen dus.
De software herkent nu de signalen afkomstig van de digit select uitgangen ,en weet dus welk digit binnen gehaald wordt.
De computer zorgmt er nu voor ,dat de afzonderlijke digits weer 'samengeplakt' worden tot een 3-digit getal.
De data komt uit de CA3162 op de pennen 2,1,15,16 ;terwijl de digit select de pennen 4,3,en 5 gebruikt. Aan de digitselect uitgangen zijn drie transistoren gekoppeld, die de displays va stroom voorzien. Wanneer we aan de collectors van deze torren nu weerstanden van 1K naar massa monteren dan kunnen we hier de digitselect pulsen afnemen (zie figuur).
De verbindingen tussen de DVM, en de PL6/7 connector zijn als volgt!

```
poort B0 (PL 6/7 pin 10) aan CA3162  pen 2  (2↑0)
poort B1 (PL 6/7 pin  9) aan CA3162  pen 1  (2↑1)
poort B2 (PL 6/7 pin  8) aan CA3162  pen 15 (2↑2)
poort B3 (PL 6/7 pin  7) aan CA3162  pen 16 (2↑3)
poort B4 (PL 6/7 pin  6) blijft open
poort B5 (PL 6/7 pin  5) digit sel. least sign. digit
poort B6 (PL 6/7 pin  4) digit sel.  next sign. digit
poort B7 (PL 6/7 pin  3) digit sel.  most sign. digit
```

PRAKTISCH:

1. de massa van de digitale meter moet worden doorverbonden met de 0V lijn van de computer.
2. De DVM kan ook uit de computer gevoed worden.
3. De door het programma gebruikte 0-page adressen zijn de volgende:

#90: waarde van het most sign. digit
#91: waarde van het next sign. digit
#93: waarde van het least sign. digit

4. De getalpresentatie (dwz. het symbool wat door het 4-bits woord weergegeven wordt):

0 t/m 9 zelfde betekenis
11 betekent '-'
12 betekent 'E' (error)

5. Voor de verbinding van de DVM met de computer kan het beste 10-aderige flatcable gebruikt worden, bij voeding uit de computer blijft er dan 1 ader ongebruikt.
6. Een nadeel: de gebruikte omzetter is niet een van de snelsten. Hou hier rekening mee in het programma bij het meten van snel varieerende grootheden.

Het idee komt oorspronkelijk uit RB nov.'81, en sept.'81

succes met de bouw, en het meten.

rudi van Drunen

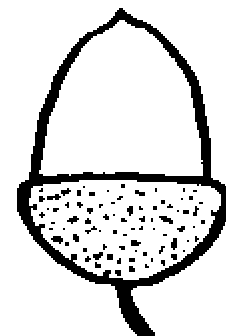
```
>LIST
    0 REM *A/D  CONVERTER TEST*
    3 REM ([redacted])
    4 REM
    5 REM R.VAN DRUNEN
    6 REM RINGOVEN 12
    7 REM 9934 LD DELFZIJL
   10 REM
   20 DIM LL6,P-1
   30[
   40:LL2 LDA@#80   B7 HOOG?
   50:LL1 BIT#B800
   60 BEQLL1          NEE? ^LL1
   70 LDA@#60        B6&B5 LAAG?
   80 BIT#B800
   90 BNELL2          NEE? ^LL2
  100 LDA#B800        HAAL WAARDE
  110 AND@#0F         MASKER SEL.
  120 STA#90          STORE MSD
  130:LL4 LDA@#40
  140:LL3 BIT#B800
  150 BEQLL3
  160 LDA@#A0
  170 BIT#B800
  180 BNELL4
  190 LDA#B800
  200 AND@#0F
  210 STA#91          STORE NSD
  220:LL6 LDA@#20
  230:LL5 BIT#B800
  240 BEQLL5
  250 LDA@#A0
  260 BIT#B800
  270 BNELL6
  280 LDA#B800
  290 AND@#0F
  300 STA#92          STORE LSU
  310 RTS             ^ BASIC
  320]
  330 P.$12;?#E1=0;@=0
  340 LINK LL2;H=U,L=U
  350 DO GOS.s;P.$30"WAARDE A/D CONV.: "U"   "
  360 IF U<L;L=U
  370 IF U>H;H=U
  380 P.'''
  390 P."LOW:   "L"    "'
  400 P."HIGH:  "H"    "
  410 U.0
  420sLINK LL2
  430 IF?#92>9 P.$7;GOS.x;R.
  440 IF?#90<10 G.t
  450 U=((?#91*10)+?#92)*-1;R.
  460tU=(?#90*100)+(?#91*10)+?#92;R.
  470xIF?#92=11;U=1000;R.
  480 U=-100;R.
  490 REM
  500 REM --------------------
  510 REM UITLEG
  520 REM 30-320 ASSEMBLER
  530 REM 330-480 HOOFD PRGR.
  540 REM s=UITLEES ROUTINE
  550 REM
  560 REM VARIABELEN
  570 REM LL     ASSEMBLER LABELS
  580 REM U      WAARDE A/D CONV.
  590 REM P      TOP
  600 REM L      MIN.WAARDE
  610 REM H      MAX. WAARDE
  620 REM --------------------
  630 REM                  RVD.
```



EEN VOORBEELD VAN EEN PROGRAMMA:

Met dit programma kan gedurende min. +/- 40 sec data van de DVM afgevraagd worden,en in een grafiek geplot worden

```
 10 REM DVM PLOTTING PROGRAM
 20t/m machine kode in ASSEMBLER
 50
100 DIM AA(768),BB(256)
110 W=0;REM GEEN TIJD TUSSEN DE METINGEN
120 FOR I=1 TO 768
130 FOR J=0 TO W;WAIT;N.
140 GOS.u
150 AAI=U
160 N.
170 FOR I=3 TO 768 S.3
180 B3(I/3)=(AA(I-2)+AA(I-1)+AA(I))/3
190 NEXT
200 M=BB1;N=M;J=1;K=1
210 FOR I=1 TO 256
220 IF BB(I)<=M M=BB(I);J=I
230 IF BB(I)>=N N=BB(I);K=I
240 NEXT
250 P.$12"MEETRESULTATEN"'';B=0
260 P."MINIMUM: "M "      NO. "J'
270 P."MAXIMUM: "N "      NO. "K'
280 FOR I=1 TO 256;BBI=%(192/999*BB(I));NEXT
290 P.'"TIK OP EEN TOETS VOOR DE GRAFIEK";LINK #FFE3
300 CLEAR 4
310 FOR I=1 TO 256;PLOT 13,I,BB(I);NEXT;P.$7;E.
500uLINK LL2
510 IF ?#92 9 THEN P.$7;GOTO 500;REM overflow
520 IF ?#90 10 GOTO550
530 U=((?#91*10)+?#92)*-1
540 R.
550 U=(?#90*100)+(?#91*10)+?#92
560 R.
```



Op r.20 staat dus:DIM LL6,P-1;FOR I=1TO 2;[: ....
Op r.50 staat dus:....;RTS;];N.
Er worden 768 metingen gedaan; wanneer in r.110 W een waarde (groter 0) krijgt wordt er tussen de metingen W*(1/60) sec. gepauseerd.
in r.170 t/m 190 wordt door middeling van 3 metingen 1 gemaakt (dit geeft een goede storings onderdrukking)
De maxima en minima worden in de regels 200 t/m 240 berekend. Ook worden de nummers van de min. en max. punten meegenomen. In r. 280 worden de waarden aangepast voor de plot-schaal ,de eigenlijke plotting gebeurt in r. 300
r.500 t/m 560 is een sbr. om de waarde binnen te halen,die de DVM op dat moment heeft, bij een overflow piept de computer en springt weer naar de mc. routine;totdat de ofl-conditie opgeheven is.

Rudi van drunen

HOE MAAK IK MIJN PROGRAMMA ONLEESBAAR

Een vrije vertaling van een artikel van John M.Nevison.
(uit het tijdschrift CREATIVE COMPUTING ; een boek van hem over dit onderwerp heet The little book of Basic Style; Addison Wesly.

De hoogtijdagen van de geheimschrift-programmeur zijn voorbij. (De naamsvorm die gebruikt wordt slaat op man of vrouw). De machinetaal, uit de jaren 50, werd vervangen door FORTRAN. In de zestiger jaren werd 't BASIC en COBOL. En in de zeventiger jaren is het 'gestructureerd programmeren' geblazen: PASCAL e.d.. Het wordt steeds moeilijker voor onze programmeur die zijn werk onleesbaar wil houden om zijn standing te bewaren. Geen wonder dat hij 's avonds verlekkerd werkt met zijn home-computer met weinig geheugenruimte; dus zo weinig mogelijk spaties in de programma-regels. Maar ook hier dreigt de toekomst: een BASIC programma kan uitstekend leesbaar worden gemaakt, DISK-BASIC geeft nog meer mogelijkheden daartoe, en geheugenruimte wordt steeds minder kostbaar. En mensen die eenmaal een goed leesbaar programma hebben gezien, eisen van hem dat alle programma's, ook de zijne, doorzichtig en begrijpelijk worden.
Om deze bedreiging het hoofd te bieden, moet hij zich aan enkele strikte regels houden.

1. Een programma is geen opstel.
Doe er alles aan om het idee de kop in te drukken, dat een voltooid programma net als een opstel een titel, een datum, de naam van de auteur, en een beginalinea over de bedoeling moet vertonen. Zonder zo'n introductie weet de lezer niet wie 't programma schreef, en waarom. Hij weet niet wat die variabelen allemaal voorstellen. De kans is groot dat hij niet de moeite zal nemen om dat allemaal uit te zoeken. Prettig, want niemand zal je programma zo kunnen onderzoeken en criticeren!

2. Commentaar is overbodig.
Zelfs als je de openingsfrasen eraf hebt gehaald, zullen die commentaar zinnen (REM.....) nog teveel structuur in je programma zichtbaar houden.Gooi ze er zonder dralen uit.
REM betekend REMOVE. Geef de lezer geen kans.
Pas op voor PRINT-opdrachten, die zouden kunnen verraden wat dat programma aan 't uitvoeren is!

3. Bederf de ogen.
Normale schrijftaal heeft allerlei trucjes om tekst leesbaar te maken. De geheime programmeur moetbeslistgeenspatiesgebruikenalshijhetmaareventanvoorkomen.
Inspringen m.b.v. spaties kan de lus onthullen. Weg ermee. Zo worden die lussen bijna onvindbaar. De logische structuur van het programma is nu practisch geheel verloren.De doorzettende lezer moet de regels een voor een doorworstelen en langzaam proberen er achter te komen hoe 't programma in elkaar zit.En dan heeft onze BASIC-versie voorzien van schitterende afkortingen dus afkorten wat er maar af te korten valt.
Nu moet de lezer letter voor letter doorworstelen en dat houdt niemand lang vol. De privacy van je programma is nu volledig verzekerd.

4. Gooi de logische gang van zaken door de war.
Dat gaat vanzelf al als je je aan de eerste drie regels houdt ! Toch moet je je programma nog even controleren. Een klein beetje extra werk kan geen kwaad. Vervang de FOR...NEXT loops door IF...THEN....GOTO. Het enige wat je programma nog gemeen heeft met het originele programma, dat is de uitvoer.
Heel weinig lezers zullen daar achter komen. De uiteindelijke test voor het ware geheim is dat zoiets eerst gedraaid moet worden om te "weten" wat het doet. Een gevolg van deze regels: De geheime programmeur zal de meeste voldoening smaken wanneer hij na enige tijd zelf zijn oude programma's niet meer begrijpt. Want als hij ZELF dat niet meer kan, dan zal niemand het kunnen !

INDEX

Het programma INDEX is een variant op het in A.N.3-83 beschreven cassette-catalog programma. Het huidige programma geeft alleen de naam, begin- ,eind- en startadres van elke file op de band weer, naamloze files worden overgeslagen.

```
0 REM ******** Index ******** (C) Eric Sprangers
1 DIM II6,SS2;F.I=T.TOT.+40S.4;'I=#FFFF;N.
2 F.I=1 TO 2;P.$21;P=#2800;[
3:II0 JSR #F844;:II1 JSR SS0;:II4 JSR #FBE2:LDA #D9;BNE II1
4LDX @#FF;:II3 INX;LDA #ED,X;STA #140,X;CMP @#0D;BNE II3
5BEQ II6::II5 JSR SS0;BNE II4;JSR#FBE2;:II6 LDA #DB;BMI II5
6JSR #F992;LDA #D5;SBC #D9;STA #D1;LDA #D4;STA #D0:LDX @#D0
7JSR #F7F1;CLC;LDA #D8:ADC #D4;STA #D4;LDA #D5;ADC @#00
8STA #D5:JSR #F7EC;JSR #FFED;BNE II1::SS0 JSR #FBBE:BVS SS1
9PLA;PLA;RTS;:SS1 BNE SS0;JMP #FBC9
10];N.;P.$6;IN."naam band"$P;?#FE=0
11 REM printer-besturing voor Microline
12 P.$2$27$66$31'''$P''$30$3$12:?#FE=10;LI.#FC3E
13 P.$2;LI.II0;P.$3;E.
```

TWEEDE 16K RAM KAART VERVANGT LAGE GEHEUGEN

WAAROM DEZE TWEEDE 16K KAART? SIMPELWEG OM DE TOEGENOMEN BELASTING VAN ADRES- EN DATALIJNEN, MAAR OOK VAN DE VOEDING, TE BEPERKEN.
UIT DE OVERVOLLE KAST VAN DE ATOM VERDWIJNEN ZO'N 16 IC'S. IN PLAATS DAARVAN KOMT WEL DE TWEEDE 16K KAART, MAAR DEZE VRAAGT VEEL MINDER STROOM. BOVENDIEN DE MOGELIJKHEDEN ZOALS WRITE PROTECT EN BATTERY BACKUP. ER ZIJN ENKELE KLEINE WIJZIGINGEN NODIG OP DE CLUB 16K KAART. NU DE TECHNISCHE KANT VAN DE ZAAK.

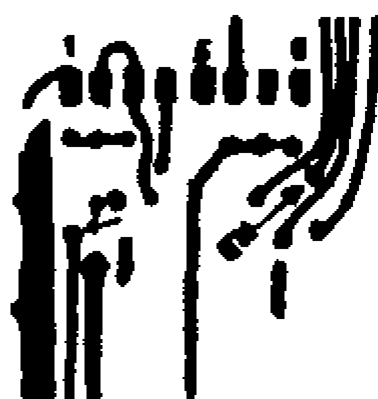
VERWACHT WORDT DAT U AL IN HET BEZIT BENT VAN EEN 16K KAART OP HEX 4000-7FFF. ZO NIET, DAN KUNT U DE GEWIJZIGDE KAART NIET ZONDER MEER GEBRUIKEN.
DE BOVENKAART (ZO NOEM IK NU DEZE TWEEDE KAART) GEBRUIKT NAMELIJK DE BUFFERS EN WAT LOGICA VAN DE ONDERKAART. DE BOVENKAART WORDT D.M.V. EEN 10-POLIGE EN DRIE 8-POLIGE CONNECTOREN MET DE ONDERKAART VERBONDEN.
ZELF GEBRUIK IK HIERVOOR AFGEZAAGDE IC VOETJES MET TUSSEN BEIDE PRINTEN KONTAKT-PENNEN. DE CONNECTOREN OP DE BOVENKAART BEVINDEN ZICH AAN DE SOLDEERZIJDE.

VOLG NU DE AANWIJZINGEN PUNT VOOR PUNT:
1. PLAATS DE VERBINDINGSCONNECTOREN OP BEIDE KAARTEN
2. PLAATS DE IC VOETEN BEHALVE VAN IC9-10-11-12
3. PLAATS C1 T/M C17, R1 EN R22 T/M R27
4. VERBIND PIN 11 EN 12 VAN IC13 MET PIN 12 EN 13 VAN IC12
5. VERBIND OP IC13 PIN 13 MET PIN 9
6. PLAATS EEN WEERSTAND VAN 2K2 OP IC12 TUSSEN PIN 11 EN PIN 14
7. VERBIND OP IC13 PIN 1 MET PIN 8
8. ONDERBREEK A14 BIJ PIN 6 VAN IC14 EN VERBIND A14 (DEZE KOMT VAN IC11 PIN 3) MET IC13 PIN 2 EN PIN 3
9. PLAATS HET SCHAKELAARTJE VAN DE "WRITE PROTECT" AL DAN NIET OP EEN IC VOETJE

VOLG VERDER DE AANWIJZINGEN DIE U KREEG BIJ DE PRINT.

DE VOLGENDE WIJZIGINGEN OP BEIDE KAARTEN UITVOEREN (ZIE OOK DE TEKENING):
1. VERBIND "PD" MET 5 VAN DE CONNECTOR BIJ DE BATTERIJ
2. VERBIND "EN" MET 8 VAN DE CONNECTOR BIJ DE BATTERIJ

PUNT 1 = NRDS
2 = AARDE
3 = (ENABLE)
4 = NWDS
5 = POWER DOWN
6 = + 11
7 =
8 = ENABLE



PD = HET SMALLE PRINTSPOOR RECHTS VAN C10
EN = HET DERDE PRINTSPOOR VAN LINKS

DENKT U HIERVOOR HET PRINCIPE SCHEMA VAN DE 16K RAM KAART NODIG TE HEBBEN, DAN IS DAT VERKRIJGBAAR BIJ DE REGIO ARNHEM E.O.
DE ONDERKAART BLIJFT NA DE WIJZIGINGEN UITWISSELBAAR MET DE ONGEWIJZIGDE KAART.

NU DE WIJZIGINGEN IN DE ATOM ZELF:
1. VERWIJDER IC10 T/M IC19 (OOK DE VARKENS), IC51, IC52, IC6, IC7
2. VERBIND OP VOETJE IC6 DE PINNEN 10, 11, 12, 13, 9, 16
3. HAAL IC5 UIT Z'N VOET, BUIG PIN 12 NAAR BUITEN, VERBIND DEZE MET PIN 14 EN STEEK TERUG IN DE VOET

ALLES NU CONTROLEREN, AANSLUITEN EN TESTEN. BIJ MIJ GEBEURDE ER NIETS.
DOOR (VOOR MIJ) ONVERKLAARBARE OORZAAK WERD ER NIETS OF VERKEERD IN DE ZEROPAGE GESCHREVEN. ZONDER IC4 IN DE ATOM (WERD TIJDELIJK DOORVERBONDEN) WERKTE ALLES PRIMA. DOOR DE VOLGENDE WIJZIGINGEN IN DE ATOM WERD DIT OPGELOST:

1. HAAL IC4 UIT Z'N VOET. BUIG PIN 11 NAAR BUITEN EN STEEK TERUG
2. VERBIND IC4 PIN 11 MET IC8 PIN 11 (IS HET PRINTSPOOR RECHTS VAN OPDRUK IC7)
3. STEEK EEN DRAADBRUGGETJE IN VOETJE IC7 VAN PIN 1 NAAR PIN 4 EN VERBIND DEZE MET DE R/W LIJN (IS HET PRINTSPOORTJE RECHTS VAN OPDRUK IC8)

PIN 11 VAN IC4 KOMT DUS AAN EEN GEINVERTEERDE R/W.
ALLES WERKT NU NAAR BEHOREN.

NOG EEN MOGELIJKHEID:
DE VRIJE INGANGEN OP IC5 VAN DE ATOM ZIJN TE GEBRUIKEN OM BIJV. HEX A000 OF EEN ANDER ADRES BUITEN DE BUS TE BRENGEN; WIJZIG HIERTOE OP DE ONDERKAART:

1. MAAK PIN 4 VAN IC13 LOS EN VERBIND DEZE MET PIN 11 VAN IC12
2. VERBIND OP IC13 PIN 5 EN PIN 6 VIA EEN WEERSTAND VAN 2K2 MET PIN 7
3. MAAK PIN 5 EN PIN 6 VAN IC13 LOS VAN DE CONNECTOR BIJ DE BATTERIJ (CONNECTORPUNT 3, ZIE DE TEKENING) EN VOER DEZE NAAR BUITEN

U HEBT NU EEN LIJN WAARMEE DE ONDERKAART (HARD OF SOFT) GESCHAKELD KAN WORDEN. LAAG=AAN EN HOOG=UIT.

U KUNT MEERDERE ONDERKAARTEN PARALLEL SCHAKELEN ,DEGENE WAARVAN DEZE LIJN LAAG IS STAAT AAN. DE BOVENKAART BLIJFT RUSTIG DOOR GAAN, OOK AL STAAT DE ONDERKAART WAAR HIJ OP ZIT, UIT. BOVENDIEN BLIJFT DE ONDERKAART NOG STEEDS UNIVERSEEL.

TENSLOTTE ZOU IK NOG WILLEN VRAGEN: WIE HEEFT ER VOOR MIJ DE SOFTWARE WAAR IK AL ZO LANG NAAR ZOEK, NL. MORSE EN RTTY ZENDEN EN ONTVANGEN, EVENTUEEL VIA DE RS232 OP HEX B400.

SUCCES MET DE BOUW' AP PETERS REGIO ARNHEM E.O.

16 kByte van #0000 tot #4000

Met enkele wijzigingen is de 16-kB clubkaart geschikt te maken voor het geheugengebied van #0000 t/m #3FFF. Dit heeft dan als resultaat dat het hele geheugen tot aan de video-RAM gevuld is met cMOS, zodat de voeding in ieder geval iets minder warm wordt. Verder heeft een ieder vrijelijk de beschikking over meer geheugen (vanaf #0400). Raadpleeg uw memory-map.

Wat te doen:

A. Moederboard.
1. Alle 2114's verwijderen behalve die van het video-gedeelte. Verwijder dus de IC's 10 t/m 19, 51 en 52.
2. Poot 12 van IC 5 (74LS30) uit de socket verwijderen. Om op save te spelen (het is dus niet direct noodzakelijk) kan dit pootje tesamen met de andere vrije aansluitingen (vrijgekomen door het verwijderen van de RAM; poten 2,3,4,5,6 en 12) aan de +5V gelegd worden via een weerstand van 1-10 kohm.
3. Het IC 6 (74LS138) verwijderen.

B. 16-k kaart.
1. DOORKRASSEN (voorzichtig) van het spoor bij poot 6 van IC 14 (74LS138) aan de SOLDEERZIJDE.
2. Het losgehaalde spoor (NIET de kant van IC 14, maar de kant die uiteindelijk naar IC 11 (74LS244) loopt) verbinden met de poten 11 en 12 van IC 13 (74LS01).
3. Poot 6 van IC 14 (juist ja, dat ligt aan de andere kant van de eerder gemaakte kras) verbinden met poot 13 van IC 13.
4. Weerstand van 1-10 kohm leggen van datzelfde pootje 13 van IC 13 naar de +5V.

De kaart is nu gereed om aangesloten te worden.Dit kan op verschillende manieren:
- In de kast gestapeld op evt. de reeds aanwezige geheugen- en schakelkaart.
- Buiten de kast op een zgn. backplane.
- Elke andere zelf uit te puzzelen wijze.

DISKGEBRUIKERS OPGEPAST.
Het controller-IC, de 8271, maakt gebruik van de geheugenplaatsen #0A00 t/m #0A7F en #2000 t/m #27FF (bij mij althans). Er bestaat nu een dubbel geheugenbestand op die adressen. Hier zal dus het een en ander moeten gebeuren om stoute dingen te voorkomen. Wie daar een goed idee over heeft, laat die het mij even weten?

Succes, Gerhard Visser.
(naar een idee van CdR)

Interupts

Voordat we over interupts gaan praten, moeten we eerst duidelijk vaststellen wat een interupt is. Voor velen is het een gevleugeld, duister begrip. Voor anderen de schrik van tentamens. (!) Collega auteur D. van Dale vertelt erover: 'in de rede vallen, onderbreken'.
Een interupt voorbeeld: U zit in uw Acorn Nieuws te lezen. Uw vrouw komt binnen en zegt 'wil je koffie ?'. Het lopende proces (A.N. lezen), en een ander proces (wil ik inderdaad koffie) wordt afgehandeld. Is dit klaar, dan wordt weer verdergegaan met het lopende proces (het lezen).
Op exact gelijke wijze werkt een processor een interupt aanvrage af. De hoofdrol is een hoofdprogramma. Dit loopt b.v. in een lus. De onderbreking van zo'n programma kan alleen door een interupt gebeuren. Komt er een interupt, dan wordt naar een ander deel van een programma gesprongen.
Als het u nog niet duidelijk is, dan een volgend voorbeeld. (Het doet de waarheid een klein beetje geweld aan, maar het verduidelijkt.) Stel, we hebben het volgende (onzin)programma:

```
      P=#2800:[
        LDA #70
        STA #70
        JMP #2800
```

Het is duidelijk een gesloten lus. Nu wordt er een schakelaar (als voorbeeld) omgehaald. Door dit schakelaartje, wordt nu naar een subroutine gesprongen, deze wordt afgewerkt en keert nu weer terug naar waar deze was. Het is dus net of de stand van de schakelaar bepaald of de subroutine wel of niet gedaan wordt.

Er zijn drie vormen van interupt:

- I/O interupt
- geprogrammeerde interupt
- DMA interupt (Direkt Memory Acces)

Allereerst: I/O interupt

Het specifieke van I/O interupt is, dan het lopende programma terstond wordt onderbroken en met de interupt subroutine verder gaat. (De instruktie waar de processor mee bezig is wordt wel eerst algemaakt)
I/O interupt heeft altijd te maken met 2 signaal ingangen op de 6502 processor n.l. NMI en IRQ. Deze twee ingangen funktioneren net als de vrouw in het genoemde voorbeeld.
Er zijn in totaal 3 ingangen die het lopende programma kunnen onderbreken. In volgorde:

1 RESET

Dit is een bekende. In de Atom is deze verbonden met de 'BREAK' toets. De reset reageert op de opgaande flank. Van '0' naar '1' dus. Het lopende programma wordt altijd onderbroken en er wordt een JMP (#FFFC) uitgevoerd. Bij iedere 6502 computer is dit zo. Een JMP (#FFFC) is, zoals al eens uitgelegd, een indirekte jump. In dit geval wordt de data van #FFFC en #FFFD geladen (resp. #3F en #FF) en er wordt naar #FF3F gesprongen. Probeer het maar eens LINK #FF3F. Het geeft het zelfde resultaat als een 'break' Een reset start een computer op. In de Acorn worden eerst alle vectoren (COS e.d.) geplaatst in RAM #00-#3FF en daarna vertelt deze dat hij Acorn Atom heet. Dan wordt de stackpointer op #FF gezet (met een schone lei beginnen) en daarna is de computer gereed voor gegeveninvoer.

2 IRQ

Dit is de Interupt ReQuest. Dit is een beleefde vorm van interupt. Aan de hand van een nog niet besproken vlag uit het P register, wordt bepaald of de interupt aanvrage wel of niet geaccepteerd wordt. Deze IRQ is aktief als het niveau '0' is of als het '0' wordt. Ook hier is sprake van een indirekte jump voor het afwerken van de interupt. Er zit een truucje achter die voor u als cusisten niet zo van belang is. Het is voldoende om te weten dant het een JMP(#204) is. Dit is een RAM locatie en is dus door u als programmeur te veranderen.

3 NMI

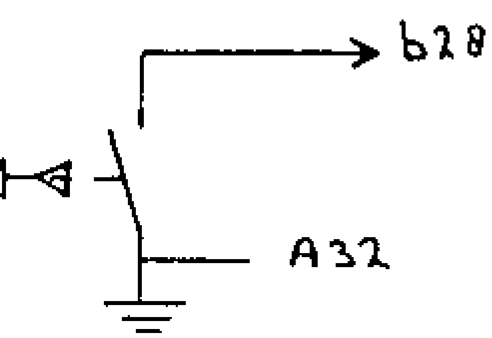
Dit is de Niet Maskeerbare Interupt. Deze interupt reageert op de achterflank van een signaal. Op de '1' naar '0' overgang dus. Komt er een '1' - '0' overgang dan maakt de processor zijn instruktie af en doet een JMP (#200). Bij de Atom althans. Ook hier is dus door de programmeur te bepalen waar de NMI heen springt. De locaties #200 en #201 wijzen naar deze plaats. Door een reset van de 6502 (een 'BREAK' dus) wordt de inhoud van deze twee adressen niet veranderd, dit in tegenstelling tot de IRQ vector die na iedere reset weer op #A000 staat. Tot zover iets over de hardware interupts. Nu iets over de software interupt.

BRK

Zodra ergens in een programma een BRK (code #00) wordt uitgevoerd, dan treedt een IRQ situatie op. In de ROM wordt nu gekeken of het een IRQ danwel een BRK is. Is het namelijk een BRK dan is de B vlag uit het status register P een '1'. Als deze inderdaad 1 is dan wordt er in de Atom een JMP(#202) gedaan. Ook deze vector is door de gebruiker naar believen te wijzigen.

AAN DE SLAG MET DE IRQ

Voorbeelden voor interuptprogramma's zijn vaak erg moeilijk. Toch heb ik geprobeerd iets eenvoudigs te bedenken. Bewust laat ik de interupts met de VIA 6522 liggen, anders zou deze aflevering nog erger worden dan dat deze al is.
Allereerst wat hardware in deze software cursus. Het is de bedoeling dat met een eenvoudig drukschakelaar een interupt gegenereed gaat worden. De schakelaar moet als volgt worden aangesloten.



Dit is alle hardware die nodig is. Het valt nog al mee dus. Met deze schakeling is het nu mogelijk een interupt op te wekken. En nu een stukje software ondersteuning. Als de schakelaar bedient wordt, wordt het scherm schoon gemaakt net als CONTROL L dit doet. We moeten ons bedenken dat het lopende programma wordt onderbroken, dus zorg ervoor dat als de interupt subroutine afgewerkt is, de inhoud van A X en Y precies zo is als dat deze waren voor de ISR (Interupt SubRoutine).
Als er een interupt op de IRQ komt, gebeurt er het volgende:
* Eerst wordt gekeken hoe de I vlag uit het status register er bij staat. Is deze '1' dan komt er geen interupt.
* Is deze I vlag '0' ,dan wel. Eerst wordt het P register op de stack gezet en daarna de Accu. Dit gebeurt altijd en daar hoeft de programmeur dus geen rekening mee te houden. Hij moet alleen wel zorgen dat de Accu er wel weer wordt afgehaald.
* Zodra de ISR begint MOET de I vlag '1' gemaakt worden, anders zou er een interupt bij komen, omdat schakelaars nogal denderen. (Dit is overgens ook de reden waarom ik niet de NMI als voorbeeld genomen heb. Deze is immers niet maskeerbaar.) Het '1' maken van de I vlag gebeurt met SEI (SEt I flag). Het '0' maken gebeurd met CLI (CLear I flag).

Het ISR deel van het programma zou er zo uit kunnen zien.

```
   SEI          MAAK I VLAG 1
   TXA          X NAAR ACCU
   PHA          ACCU NAAR STACK
   TYA          Y NAAR ACCU
   PHA          ACCU NAAR STACK
   JSR #FD69    =P.$12
   PLA          ACCU VAN STACK
   TAY          ACCU NAAR Y
   PLA          ACCU VAN STACK
   TAX          ACCU NAAR X
   PLA          ACCU VAN STACK
   CLI          MAAK I VLAG 0
   RTI          RETURN VAN INTERUPT
```

De RTI lijkt ers op de ons bekende RTS. De RTI haalt eerst het P resister van de stack en keert dan terus van waar hij kwam. RTI is dus het zelfde als PLP:RTS.
De ISR is nu een feit. Het enise wat nu nos moet sebeuren is #204 en #205 wijzisen, (deze wijzen immers naar de ISR) en de I vlas resetten.

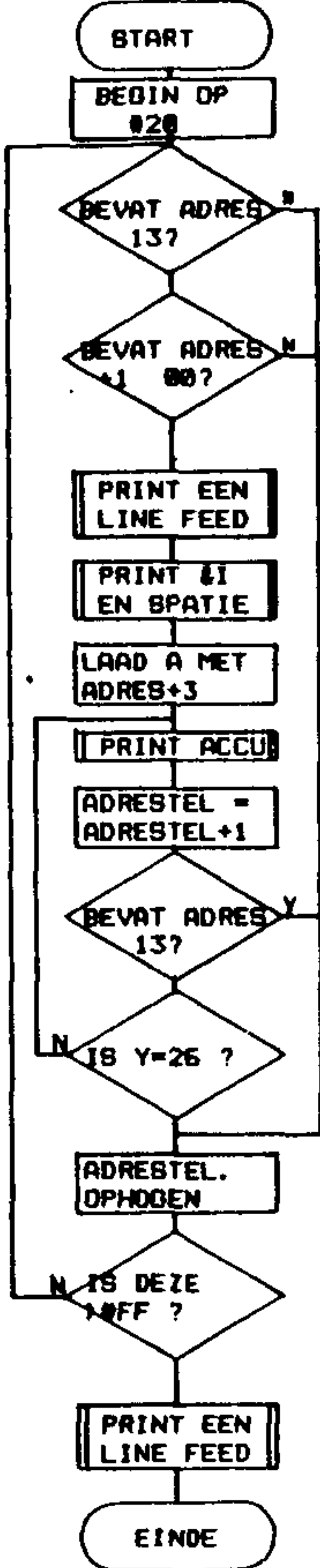
```
   LDA @#28     HIGH BYTE
   STA  #205    STORE
   LDA @#00     LOW  BYTE
   STA  #204    STORE
   CLI          CLEAR I VLAG
```

Alles bij elkaar zou het er zo uit kunnen zien.

```
 10 REM P.$12 MET IRQ
 20 DIM LL0;P=#2800
 30   [:SEI
 40     TXA
 50     PHA
 60     TYA
 70     PHA
 80     JSR #FD69  P.$12
 90     PLA
100     TAY
110     PLA
120     TAX
130     PLA
140     CLI
150     RTI  EINDE ISR
160:LL0 LDA @#28
170     STA  #205
180     LDA @#00
190     STA  #204
200     CLI
210     RTS
220 ]
230 LINK LL0
240 END
```

Probeer na het se'run'ed te hebben eens: A=0:DO:A=A+1:PRINT A':UNTIL 0 {retun} Druk eens op de IRQ schakelaar. Mooi, he !
Een laatste opmerkins: de toolbox en de Josbox maken sebruik van de IRQ die de VIA 6522 afseeft. Met de Josbox kan niet door bovenstaand prosramma worden se'step'ed. De monitor heeft er seen moeite mee, omdat deze niet de IRQ nodis heeft. Ook een eventueel aanseloten printer kan wat moeilijkheden hebben met het prosramma, omdat deze ook op een vorm van interupt werkt, namelijk de seprosrammeerde interupt. In deel 9 komt deze vorm uitvoeris aan de orde.

Leendert Bijnaste




```
10REM CATALOG
20DIM LL4
30FOR N=0 TO 4
40 LLN=-1
50NEXT N
60FOR N=0 TO 1
70 P=#2800:[
80       LDA   @#20   \ of @#8
90       STA    #71
100:LL0   LDY   @#00
110       STY    #70
120       LDA   (#70),Y
130       CMP   @13
140       BNE    LL3
150       INY
160       LDA   (#70),Y
170       BNE    LL3
180       JSR    #FFED
190       LDA    #71
200       JSR    #F7FA
210       LDY   @#3
220:LL1    LDA (#70),Y
250        JSR  #FFF4
260:LL2    INY
270        LDA (#70),Y
280        CMP @13
290        BEQ  LL3
300        CPY @26
310        BNE  LL1
320 :LL3 INC    #71
330       BNE   LL0
340      JMP    #FFED
350]
360NEXT N
370LINK #2800
380END
```

!--**--Leendert Bijnagte--**--!

Hier volgt een beschrijving van het programma zoals beschreven in de vorige Acorn Nieuws. Allereerst beginnen we met een analyse op te zetten. Dit is altijd een goede gewoonte. Het lijkt mischien iets meer werk, doch het geeft een inzicht in de werkmethode. Het geeft het programma een effectieve wending. Tevens zorgt het voor een stuk documentatie en is een leidraad bij het fouten opsporen. Ook bij proffesionele bedrijven wordt gemiddeld zo'n 30 % van de programmeertijd besteed aan analyse en documentatie.
Dit keer een iets minder uitgebreide beschrijving, want het moet nu al iets duidelijker zijn. Bestudeer het stroomdiagram van de bijlage maar eens. De beroemde post indexed indirekt wordt weer van stal gehaald. Voor de rest is het diagram erg rechtlijnig. Er zitten 3 ROM routines in. De eerste (JSR #FFED) is de machinetaal vervanger van PRINT'. Nummer 2 (JSR #F7FA) print de waarde die in de accu staat als hexa-decimaal getal en print hierna een spatie. JSR #FFF4 tenslotte is een routine die de inhoud van de accu als een ASCII karakter uitprint. Voor de rest zitten er geen moeilijkheden in.
Wat dit programma doet? Wel het print de eerste regel van iedere textspace waar een basic programma staat, tot de return (13) of tot het 26ste teken. Als u dus op #2900 en op #8200 een programma heeft staan, wordt dus de eerste regel gelist.
voobeeld:
LINK #2000

29 REM GETAL PI
82 REM FUNKTIE INTERGREREN

Het kan gebeuren dat er op een vreemde plaats een 13 en een 00 staat. Dit kan gebeuren als de computer net aangeschakeld wordt. Er staan dan allerlei wilde getallen in. Als inderdaad op b.v. #8800 een 13 staat en op #8801 een 00 dan 'ziet' het programma dit als een basic textspace, zonder dat dit er een is en gaat dus ook de getallen achter #8801 als ASCII waarden uitprinten. Dit stelt uiteraard niets zinnigs voor. Het kan alleen maar problemen geven omdat b.v. het beeldscherm wordt uitgezet of de printer aan. In het basic programma zou het opgelost worden door deze regel er bij te voegen:

```
65 IF A?J<32 THEN GOTO 80
```

In de assembler uitvoering zou het dan worden:

```
230   CMP @32
240   BCC LL3
```

Het eventueel aanwezig zijn van een 0D en een 00 is alleen van belang voor de plaatsen #**00 en #**01 (** is een getal tussen de 00 en de #FF). Andere plaatsen worden toch niet bekeken.

Dit programma kan uitstekend funktioneren als extra statement. Bij mij heet het CAT.
succes !!

P.E.R.T.

Eerste aflevering.

INLEIDING

PERT staat voor Performance Evaluation and Review Technique, een methode om ingewikkelde activiteiten te plannen.
Vooral nieuwe activiteiten, waarvoor nog geen of slechts weinig betrouwbare schattingen van de benodigde tijdsduur bekend zijn, kunnen goed met behulp van PERT geschat, gepland en gecontroleerd worden.

De methode werd ontwikkeld door de Amerikaanse marine bij de ontwikkeling van de Polaris-raket. Deze moest in een zeer kort tijdsbestek gerealiseerd worden, maar er bestond grote onzekerheid aangaande de benodigde tijdsduur voor de deelactiviteiten, waarin het gehele project opgedeeld kon worden. Bovendien was de start van bepaalde deelactiviteiten afhankelijk van de succesvolle voltooiing van eerdere deelactiviteiten.

Bij PERT heeft men dus te maken met activiteiten waarin een project verdeeld kan worden. Elke activiteit kost tijd. De activiteit strekt zich uit tussen twee mijlpalen, een beginmijlpaal (i) en een eindmijlpaal (j). Het project moet één beginpunt hebben dat tevens het laagste mijlpaal-nummer heeft en één eindpunt dat het hoogste mijlpaal-nummer heeft.

Van de tijdsduur kan men drie verwachtingen uitspreken nl. de meest optimistische tijd, de normaal geschatte tijd en de meest pessimistische tijd. Aan de hand van deze drie tijdsduren per activiteit, die van mijlpaal (i) naar mijlpaal (j) loopt, wordt de meest waarschijnlijke tijd en de standaarddeviatie berekend. Vervolgens wordt voor het gehele netwerk (= het gehele project) de meest waarschijnlijke tijd bepaald, benodigd om het gehele project te voltooien.

Hierbij blijkt dat bepaalde activiteiten op het zogenaamde "Kritieke Pad" liggen, d.w.z. deze activiteiten mogen niet later beginnen dan gepland en vertraging van deze activiteiten leidt tot een vertraging van het gehele project. Het Kritieke Pad is dus díe weg door het netwerk waar geen speling mogelijk is.

Tot slot kan men nog opgeven wat de gewenste tijd is voor realisatie van het project. Aan de hand van de gewenste tijd en de meest waarschijnlijke tijd, wordt de kans berekend dat het project klaar komt binnen de gewenste tijd.

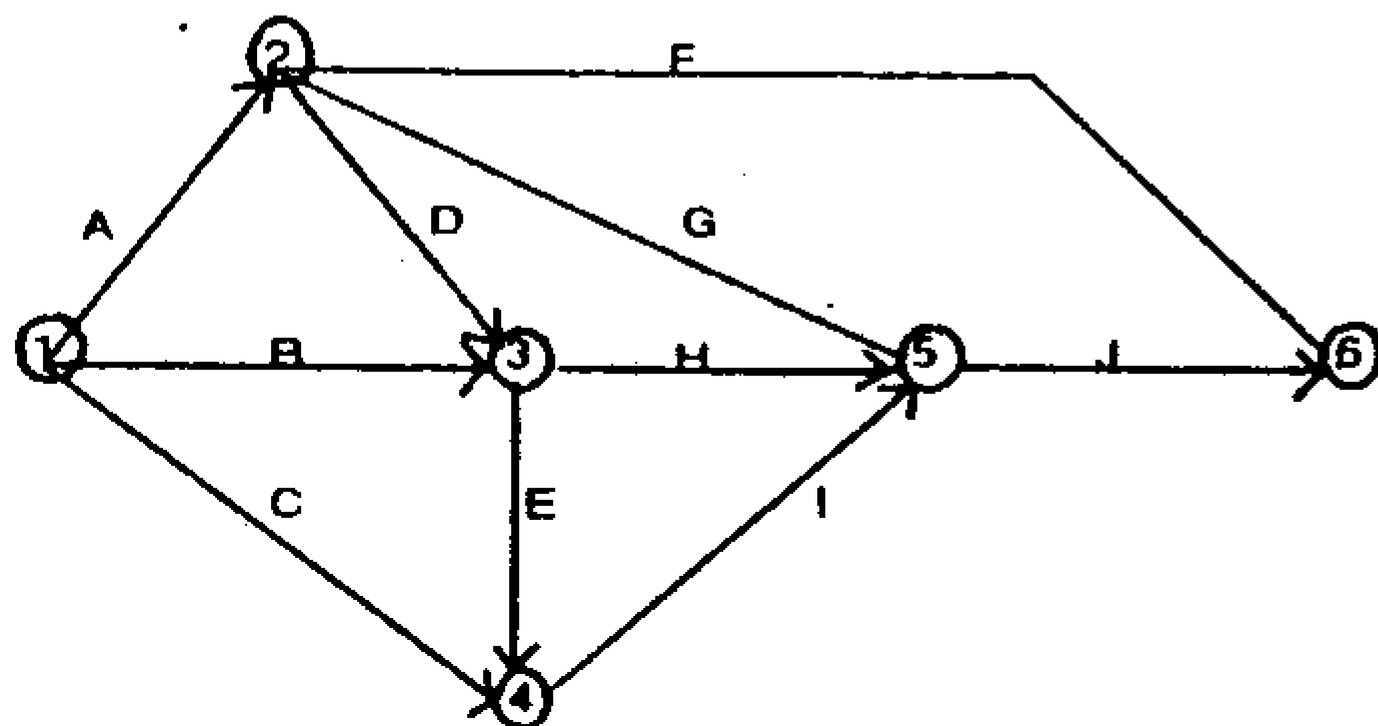
HET PROGRAMMA

Het programma beslaat 5552 B en gebruikt FP. Het is zo goed mogelijk tegen allerlei vormen van verkeerde invoer beschermd, zelfs tegen een "onlogisch" netwerk.
De basis van het programma wordt gevormd door een listing uit het boek "Basic programma's voor school en bedrijf" van M. Th. A.M. Vijftigschild, uitgegeven bij Kluwer. Het programma is aangepast naar Atom-Basic en naar uitvoer via het scherm.
Ook "Byte" van mei 1982 bevat een artikel over PERT.

Het programma kan maximaal 25 activiteiten aan. Mocht er meer benodigd zijn, verander dan de arrays %AA (4 x 25), %BB (4 x 25), %CC (3 x 25), AA (4 x 25) en SS (2 x (25 + 1)), alsmede de regels 410, 590 en 630.

Naast de invoer-routine bevat het programma een mijlpalen uitdraai, een activiteitenuitdraai, de kans dat een gewenste doorlooptijd gehaald wordt, alsmede een wijzigingsroutine, met behulp waarvan het netwerk gewijzigd of uitgebreid kan worden. Op ieder moment kan men een stuk uitvoer overslaan en naar een ander onderdeel van de uitvoer springen. De waarschijnlijkheid dat een gewenste doorlooptijd gehaald wordt, wordt echter alleen dán goed berekend, indien de gehele activiteiten uitdraai langsgelopen wordt. Bij de invoer hoeven de deelactiviteiten niet in de goede volgorde ingevoerd te worden. Het programma bevat een sorteerroutine om dit te doen.

Voorbeeld



In bovenstaand netwerk staan de mijlpalen 1 tot en met 6 afgebeeld als cirkels.

Voor de activiteiten A t/m J werden de volgende tijdsduren geschat:

		Tijdsduur (weken)		
Activiteit	Mijlpaal	optimistisch	realistisch	pessimistisch
A	1 - 2	1	2	2
B	1 - 3	2.1	4.5	6.3
C	1 - 4	4	4	4
D	2 - 3	2	4	5
E	3 - 4	3	3	5
F	2 - 6	0	1	2
G	2 - 5	1.4	2.2	6.7
H	3 - 5	2	4	6
I	4 - 5	0	0	0
J	5 - 6	2	3	4

I is een zogenaamde "dummy"-activiteit, toegevoegd om het netwerk één eindpunt te laten hebben.

Bij de uitvoering van het programma zal blijken dat de mijlpalen 1, 2, 3, 5 en 6 op het kritieke pad liggen. Het kritieke pad bestaat uit de activiteiten A, D, H en J. De verwachte doorlooptijd is 12.67 ± 0.91. Bij een gewenste doorlooptijd van 12.5 is de kans op realisatie 42.8%.

Peter Ruifrok

```
 10FDIM%AA100,%BB100,%CC75,%MM1,%YY3
 20DIMAA100,SS52,D64,F4;G.390
 30REM ***WAARSCHIJNLIJKHEID***
 40%M=ABS(%E/1.41421);%MM1=%M*(2.765672E-4 + %M*4.30638E-5)
 50%MM1=%M*(9.2705272E-4 + %M*(1.520143E-4 + %MM1))
 60%MM1=%M*(.070523078 + %M*(.0422820123 + %MM1))
 70%MM1=1-1/((1+%MM1)^16);%W=(1+%MM1)/2
 80FIF%E>0 G.100
 90%W=(1-%MM1)/2
100%W=%W*100;R.
110aF=0;A=1;B=1+G
120C=A+((B-A)/2);FIF%P>%BB(C*4+1) G.160
130FIF%P=%BB(C*4+1) G.190
140IFB=C G.180
150B=C;G.120
160IFA=C G.180
170A=C;G.120
180F=1
190R.
200bP."SORTEREN"';F.J=1TO2
210REM***INDEXEN INITIALISEREN***
220F.I=1TON;SS(I*2+J)=I;N.
230REM*SORTEREN*
240F.I=2TON;A=1;B=I;P?3=SS(I*2+J)
250C=A+((B-A)/2);P?4=SS(C*2+J)
260FIF%AA(P?3*4+J)<%AA(P?4*4+J) G.290
270FIF%AA(P?3*4+J)>%AA(P?4*4+J) G.310
280B=C;G.340
290IFB=C G.340
300B=C;G.250
310IFA=C G.340
320A=C;G.250
330REM***WISSELEN***
340IFI=B G.360
350F.X=I TO B+1 S.-1;SS(X*2+J)=SS((X-1)*2+J);N.
360SS(B*2+J)=P?3;N.;N.;R.
370cP.$12;P."**PERT**VERWACHTE KORTSTE"'
380P."DOORLOOPTIJD: ";FP.%B';P."PROJECT:"$D''';R.
390P.$12;P."***PERT METHODE***"''
400INP."WAT IS DE NAAM VAN HET PROJECT"'$D
410F.X=1TO25
420P.'"REGEL"X"          STOP=0,0"'
430FINP."MIJLPAAL(I)"%AA(X*4+1)
440FINP."MIJLPAAL(J)"%AA(X*4+2)
450FIF%AA(X*4+1)=0;FIF%AA(X*4+2)=0;G.590
460REM ***SCREENING OP FOUTEN***
470FIF%AA(X*4+1)<%AA(X*4+2) G.490
480P."FOUT *MIJLPAAL(I)>MIJLPAAL(J)*"';G.420
490P."REGEL"X
500FINP.'"KORTSTE TIJD"%CC(X*4+1)
510FINP."NORMALE TIJD"%CC(X*4+2)
520FINP."LANGSTE TIJD"%CC(X*4+3)
530FIF%CC(X*4+1)>=0 G.550
540P."FOUT *TIJDSDUUR<0*"';G.490
550FIF%CC(X*4+1)>%CC(X*4+2) G.580
560FIF%CC(X*4+2)>%CC(X*4+3) G.580
570G.600
```

```
580 P."FOUT *SAMENHANG KLOPT NIET*"';G.490
590N=X-1;X=25
600N.
610P.$12;P."BEREKENING"'
620REM***OP NUL ZETTEN MATRIX B***
630F.I=1TO25;F.J=1TO4;%BB(I*4+J)=0;N.;N.;%S=0
640REM***VERWACHTE TIJD EN SD***
650F.I=1TON;%AA(I*4+3)=(%CC(I*4+1)+4*%CC(I*4+2)+%CC(I*4+3))/6
660REM***AFRONDEN OP TWEE DECIMALEN***
670%AA(I*4+4)=(%CC(I*4+3)-%CC(I*4+1))/6;N.;GOS.b
680P."MIJLPAALBEPALING"'
690%BB(1*4+1)=%AA(SS(1*2+1)*4+1);G=1;X=0;DO;X=X+1
700IF%BB(G*4+1)=%AA(SS(X*2+2)*4+2) G.720
710G=G+1;%BB(G*4+1)=%AA(SS(X*2+2)*4+2)
720U.X=N
730P."BEPALING KORTSTE DOORLOOPTIJD"'
740F.I=1TON;%P=%AA(SS(I*2+1)*4+1);GOS.a
750REM***INDIEN F=1 KLOPT HET NETWERK NIET***
760IFF=1 G.1580
770P?1=C;%P=%AA(SS(I*2+1)*4+2);GOS.a
780IFF=1 G.1580
790P?2=C;%B=%BB(P?1*4+2)+%AA(SS(I*2+1)*4+3)
800IF%BB(P?2*4+2)>=%B G.820
810%BB(P?2*4+2)=%B
820N.
830P."BEPALING LANGSTE DOORLOOPTIJD"'
840F.I=N TO 1 S.-1;%P=%AA(SS(I*2+2)*4+2);GOS.a
850IFF=1 G.1580
860P?1=C;%P=%AA(SS(I*2+2)*4+1);GOS.a
870IFF=1 G.1580
880P?2=C;%B=%BB(P?1*4+3)+%AA(SS(I*2+2)*4+3)
890IF%BB(P?2*4+3))>=%B G.910
900%BB(P?2*4+3)=%B
910N.
920P."BEPALING KRITIEKE PAD"'
930%B=%BB(G*4+2);F.I=1 TO G;%BB(I*4+3)=%B-%BB(I*4+3)
940IFABS(%BB(I*4+3))>1E-3 G.960
950%BB(I*4+3)=0
960%BB(I*4+4)=%BB(I*4+3)-%BB(I*4+2)
970IFABS(%BB(I*4+4))>1E-3 G.990
980%BB(I*4+4)=0
990N.
1000REM***UITVOER PERT;KRITIEKE PAD***
1010GOS.c;T=0;F.I=1 TO G
1020P."MIJLPAAL      "%BB(I*4+1)'
1030FP."VROEGSTE AANVANG  "%BB(I*4+2)'
1040FP."LAATSTE AANVANG   "%BB(I*4+3)'
1050FP."SPELING          "%BB(I*4+4)'
1060FIF%BB(I*4+4)>0.1 G.1080
1070P."***KRITIEKE PAD***"'
1080P.';T=T+1
1090IF(T<1)AND(I<G) G.1150
1100P.'"WILT U DE REST VAN DE GEGEVENS?"';INP."JA=1, NEE=0"Q
1110IFT=G G.1150
1120IFQ=0 G.1140
1130GOS.c;T=0;G.1150
1140I=G
```

```
1150N.
1160REM***MIJLPALEN UITDRAAI***
1170L=0;GOS.c;T=0;F.I=1TON;%P=%AA(I*4+1);GOS.a;P?1=C
1180%P=%AA(I*4+2);GOS.a;P?2=C
1190%YY(1)=%BB(P?1*4+2)+%AA(I*4+3)
1200%YY(2)=%BB(P?2*4+3)-%AA(I*4+3)
1210FIFABS(%YY(2))>1E-3 G.1230
1220%YY(2)=0
1230%YY(3)=%BB(P?2*4+3)-%YY(1)
1240FIFABS(%YY(3))>1E-3 G.1260
1250%YY(3)=0
1260AA(I*4+1)=%AA(I*4+1);AA(I*4+2)=%AA(I*4+2)
1270P."MIJLPAAL I-J  "AA(I*4+1),AA(I*4+2)'
1280FP."VERWACHTE TIJDSDUUR  "%AA(I*4+3)'
1290FP."VROEGSTE AANVANG     "%BB(P?1*4+2)'
1300FP."VROEGSTE EINDE        "%YY(1)'
1310FP."LAATSTE AANVANG      "%YY(2)'
1320FP."LAATSTE EINDE         "%BB(P?2*4+3)'
1330FP."SPELING   "%YY(3)
1340FIF%YY(3)>0.1 G.1380
1350FIF%AA(I*4+4)<=0 G.1370
1360%S=%AA(I*4+4)^2+%S
1370P." **KRIT. PAD**"
1380P.';T=T+1;IF(T<1)AND(I<>N) G.1440
1390P.'"WILT U DE REST VAN DE GEGEVENS?"';INP."JA=1, NEE=0"Q
1400IFQ=0 G.1430
1410IFT=N G.1440
1420GOS.c;T=0;G.1440
1430I=N
1440N.
1450REM***WAARSCHIJNLIJKHEID***
1460P.$12;P."***PERT METHODE***"'';P."NAAM:"$D'
1470FP.'"VERWACHTE DOORLOOPTIJD"'%B" +/- "SQR(%S)''
1480FINP.''"WAT IS DE GEWENSTE DOORLOOPTIJD:STOP=0"%Z
1490FIF%Z=0 G.1580
1500FIF%Z<0 G.1460
1510%E=(%Z-%B)/(SQR(%S));GOS.40
1520REM***UITVOER WAARSCHIJNLIJKHEID***
1530P.$12;P."***PERT METHODE***"''"NAAM:"$D''
1540FP."VERWACHTE DOORLOOPTIJD"'%B" +/- "SQR(%S)'
1550FP."GEWENSTE DOORLOOPTIJD"'%Z''
1560FP."WAARSCHIJNLIJKHEID IN PROCENTEN"'%W;G.1480
1570REM***WIJZIGEN MIJLPAAL-GEGEVENS***
1580P.$12;P."***PERT-METHODE***"'"WIJZIGEN"''
1590IFF<>1G.1610
1600P."HET NETWERK KLOPT NIET"'"MET MYLPAAL"%P" IS IETS MIS"''
1610P.'"WILT U WIJZIGEN IN HET NETWERK"';INP."JA=1, NEE=0"Q
1620IFQ=0 G.1900
1630P.'"WELKE REGEL WILT U WIJZIGEN,";INP.'"STOP=0"R
1640IFR=0 G.610
1650IF(R>=1)AND(R<=N+1) G.1670
1660P."FOUT OPNIEUW"';G.1630
1670IFR>N G.1740
1680N=N-1;P.'"OUDE GEGEVENS REGEL  "R'
1690P."MIJLPAAL(I)  "%AA(R*4+1)'
1700P."MIJLPAAL(J)  "%AA(R*4+2)'
1710FP."KORTSTE TIJDSDUUR  "%CC(R*4+1)'
```

```
1720FP."NORMALE TIJDSDUUR  "%CC(R*4+2)'
1730FP."LANGSTE TIJDSDUUR  "%CC(R*4+3)'
1740N=N+1;P.''
1750P."NIEUWE INVOER REGEL  "R'
1760FINP."MIJLPAAL(I)"%AA(R*4+1)
1770FINP."MIJLPAAL(J)"%AA(R*4+2)
1780FIF%AA(R*4+1)<%AA(R*4+2) G.1800
1790P."FOUT *MIJLPAAL(I)>MIJLPAAL(J)*"'';G.1750
1800P."REGEL  "R
1810FINP."KORTSTE TIJD"%CC(R*4+1)
1820FINP."NORMALE TIJD"%CC(R*4+2)
1830FINP."LANGSTE TIJD"%CC(R*4+3)
1840FIF%CC(R*4+1)<=0 G.1860
1850P.'"FOUT *TIJDSDUUR <0*"';G.1800
1860FIF%CC(R*4+1)>%CC(R*4+2) G.1890
1870FIF%CC(R*4+2)>%CC(R*4+3) G.1890
1880G.1630
1890P."FOUT *SAMENHANG KLOPT NIET*"'';G.1800
1900 END
**END**
```

P.Ruifrok

DUP

Onderstaand programma "DUP" (van DUPlicate) is speciaal bedoeld voor die eden welke zelf commando's kunnen maken en een JOSBOX of AXR1 hebben. Het statement moet met een interpretor (zoals beschreven in Acorn Nieuws april 83) worden aangeroepen. Het Y-register moet het eerstvolgend te nterpreteren karakter aanwijzen.

Het is een machinecode versie van het "tape to tape" programma van Karel v. Houten in A.N. 3-1983. Het hier beschreven programma is wat flexibeler en heeft heel wat minder geheugenruimte nodig (#A7 bytes).

Het gebruik van de statement is als volgt : het commando DUP wordt gevolgd door de naam van de te copieren file tussen quotes ("), dus DUP "Voorbeeld". Hierna wordt gevraagd of het te copieren programma met SCOS of met FCOS geladen moet worden. Antwoorden met "F" of "S".Er volgt de mededeling PLAY TAPE en de file wordt geladen. Daarna vraagt het programma of de te copieren file met SCOS of met FCOS weggeschreven moet worden waarna de mededeling RECORD TAPE op het scherm verschijnt. Het programma wordt daarna naar tape geschreven waarbij het laad- en startadres niet veranderd worden. De gebruiker kan echter de file ook reloceren door achter de naam van het programma het reloceeradres te geven, eventueel gevolgd door een nieuw startadres. Als het originele laadadres van de file "Voorbeeld" gelijk is aan #2900 zorgt de opdracht DUP "Voorbeeld" 8200 ervoor dat de copie het laadadres #8200 heeft.

Het programma laadt bij het copieren alle files op #2900 en hoger. Dit is te veranderen op regel 3 en 8.

```
 0 REM ######## dup ######## (C) Eric Sprangers
 1 DIM DD3,CC3,TTO;F.I=T.TOT.+4OS.4;!I=#FFFF;N.
 2 IN."STARTADRES CODE"Z;F.I=1TO2;P.$21;P=Z;[
 3:DDO TYA;PHA;JSR CCO;PLA;TAY;JSR #FB1B;TYA;PHA:LDA @#29
 4STA #CC;LDY @#00;STY #CB;DEY;STY #CD;JSR #F969;JSR CCO;SEC
 5LDA #CB;ADC #D8;STA #D1;LDA #CC;ADC @#00;STA #D2:PLA;TAY
 6LDX @#CB;JSR #F893;BNE DD1;LDA #D5;SEC;SBC #D9;LDX #D4
 7STA #CC;STX #CB;:DD3 LDA #D6;LDX #D7;STA #CD;STX #CE
 8:DD2 LDX @#29;LDA @#00;JSR #FAD9;JMP #C2CF;:DD1 LDX @#CD
 9JSR #F893;BEQ DD3;BNE DD2
10:CCO JSR #F7D1;];$P="FAST <F> OR SLOW <S> ?";P=P+L.P;[
11LDX @#0B;JSR #FFE6;PHA;JSR #FFED;PLA;CMP @#46;BEQ CC1
12CMP @#53;BNE CCO;LDX @#05;:CC1 LDY @#09;:CC2 LDA TTO,X
13STA #20E,Y;DEX;DEY;BPL CC3;RTS;:CC3  CPY @#05;BNE CC2
14LDY @#01;BNE CC2;:TTO ;];N.
15 REM JOSBOX-adressen van resp. SAVVEC,BGTVEC en BPTVEC
16 REM (zie blz.193 Acorn-manual)
17 P!0=#FAE5;P!2=#A56A;P! 4=#A5D6;REM Scos adressen
18 P!6=#A5E7;P!8=#A527;P!10=#A592;REM Fcos adressen
19 P.$6$12"DUP   ADRES "&DDO''"VRIJ =       "%P+12';E.
```

TUTTI-FRUTTI

Nee....geen programma voor een dolgeworden fruitautomaat of iets dergelijks, maar zo maar een titel omdat ik niet weet wat ik er anders boven zou moeten zetten.
Tutti-frutti is van alles wat. Dit artikel ook.
Vandaar........

Van wie is de lucht?
Het lijkt me goed een paar woordjes te wijden aan het overnemen van programma's.
Computerprogramma's worden niet alleen in amateurkringen, maar ook professioneel driftig gecopieerd.

Wettelijk is daar nauwelijks iets tegen te doen, want noch de auteurswet, noch de octrooiwet bieden afdoende bescherming

Het is als met een grammofoonplaat: vrijwel iedereen zet wel eens een plaat op de (cassette)band. Dat is ook niet verboden (al zal de platenindustrie dat wel betreuren), mits het gecopieerde geluid niet in het openbaar wordt gereproduceerd en mits de banden niet worden verkocht.

Toch hebben band en plaat nog enige bescherming (BUMA en STEMRA), maar waterdicht is die bescherming ook niet.

Ik lees (in het Acorn nieuws en bij programma's op de band) nog al eens: "overname toegestaan, mits met bronvermelding" of "copyright".

Dit is - gezien het voorgaande - een slag in de lucht.
Wie overneemt zonder bronvermelding is op geen enkele wijze strafbaar.

Je zou kunnen zeggen: wie zijn programma's publiceert, is ze kwijt: ze zijn geen eigendom meer van de auteur. Evenmin als je kunt zeggen: ik bezit de lucht in mijn tuin!

Hopelijk komt de w... ver met een goede bescherming van computerprogramma's. Eenvoudig zal dat echter niet zijn. Overal in de wereld doktert men aan dit probleem.

Laat dit alles - hoop ik - geen rem zijn om door te gaan met het publiceren van programma's in ons clubblad.
't Zal mij in elk geval niet weerhouden!

Iets wat me dwars zit

Op het moment dat ik dit schrijf heb ik sinds ongeveer een week enkele clubbandjes in mijn bezit, met in totaal zo'n dikke 20 programma's.

Ondanks alle verwoede pogingen ben ik er in geslaagd er slechts enkele "aan de praat" te krijgen.
Voor bijna alle programma's wordt verwezen naar nummers van het Acorn nieuws van vorig jaar, maar die heb ik niet.

Het zal uiteindelijk allemaal wel goed komen, maar er zit toch wat scheef.

Ik ben van mening dat twee zaken beslist niet uit het oog verloren mogen worden:

- niet alle computerhobbyisten zijn doorgewinterde amateurs; er zijn ook beginners;
- over enige tijd zullen de programma's nog steeds bestaan, maar zal het drukwerk moeilijk of.. helemaal niet meer verkrijgbaar zijn.

Daarom moet m.i. een programma -zeker voor amateurs - geheel "zelfverklarend" zijn. Het moet voldoende zijn als men OAD intoetst (met evt. de naam van het programma) en RUN.

Het rogramma moet dan zelf alle instructies printen om verder te kunnen gaan.

Mag ik u een voorbeeld geven?

Het programma "LICHTKRANT" van Joost de Wijs (wat een fantastisch programma!) heb ik als volgt gewijzigd:

```
1Ø P.Ø12"lichtkrant"''"EEN PROGRAMMA VAN JOOST DE WIJS."'
2Ø P."ATTENTIE!"'"LICHTKRANT ALTIJD STOPPEN
   MET              shift"''
3Ø P."WELKE LETTERGROOTTE?"''"1 GROOT"''"2 MIDDELMATIG"''
4Ø IN."3 KLEINER"'M
1ØØ GRMOD; DIMA64; ?225=Ø;P."VOER DE TEKST IN (MAX.
   64 KAR.)"'
```

Regel 490 wordt dan:

```
49Ø CLEAR M; LINK#2800
```

In het dekseltje van het cassettedoosje staat simpelweg:
LICHTKRANT, LOAD"LK"; de rest wijst zich vanzelf.

En zo doe ik dat met alle programma's die ik op de band vastleg. Voor elke gebruiker is het dan een fluitje van een cent om zo'n programma te runnen; iets anders raadplegen is volstrekt overbodig.

Het is overigens een kleine moeite alle REM regels, die betrekking hebben op de uitvoering van het programma te vervangen door PRINT regels. Daar zijn veel gebruikers bij gebaat!

De graphics.methode van Frank de Groot

Met deze methode zijn aardige letters te maken.

Maak om te beginnen twee blanco PRINT-regels (wist u eigenlijk al):

```
1Ø P.Ø12
2Ø P."                                  "'
3Ø P."                                  "; END
```

- toets nu in: ?#E7=#EE
- breng de cursor naar regel 2Ø (N.B. een spatie geef ik aan met: 𝕾)

- toets op regel 2Ø in: DD [illegible] TD 5 D 5 D 5 TD 5 D
- nu regel 3Ø: TD 5 TD 5 E 5 E 5 ED 5 L
- na BREAK, OLD en RUN een prachtig: "HALLO !"

De overige letters (lang niet allemaal een wonder van schoonheid) kunnen als volgt worden verkregen:

A =	TD TD	K =	EV DC	V =	DD GF
B =	UD ED	L =	D E	W =	D D GGF
C=	T E	M =	DGFD D D	X =	GF TD
D =	TD ED	N =	DKD D D	Y =	GD D
E =	U E	O =	TD ED	Z =	X E
F =	U D	P =	UD D		
G =	TN EY	R =	UY DC		
H =	DD TD	S =	T GD		
I =	L D	T =	ZTN D		
J =	D GD	U =	DD ED		

Zoals ik al schreef: niet allemaal wondertjes van grafische kunst, maar misschien weet iemand het beter!

VAR & ZERO

Var en zero zijn bekende commando's voor de toolkit bezitters.Maar aangezien er veel nadelen aan deze toolkit zitten,is het naar mijn mening beter om deze toolkit eruit te halen,zodat deze eprom weer voor andere doeleinden kan dienen.
Met de interpretor van Foppe Bouma of de schakelkaart is het dan mogelijk om de nog overige nuttigste commando's,die niet in de josbox zitten toch te behouden.
Twee van die commando's zijn VAR en ZERO.
NB.Voor de niet toolkit kenners:het commando "VAR" print alle variabelen (A..Z) uit op het scherm,"ZERO" maakt al deze variabelen 0.

> PROGRAMMA "VAR"-FUNCTIE

```
>L.
   20 F.X=0 TO 3 ;LLX=-1;N.
   21 IN."BASE ADRESS"Q
   30 F.X=0 TO 1
   31 P=Q
   50[LDX@1;LDY@1;LDA(#05),Y
   60 PHA;LDY@#00;STY#321;INY
   70:LL1TYA;PHA;CLC;ADC@#40;JSR#FFF4;LDA@#3D
   80 JSR#FFF4;LDX@01;JSR#C8E3;LDX@0;JSR#C589;LDA#B001;AND@#20
   90 BEQLL3:LDA@#0E;SEC;SBC#07;TAY;LDA@#20
  100 LL2JSR#FFF4;DEY
  110 BNELL2;STY#07;PLA;TAY;INY;CPY@#1B;BNELL1
  120:LL3PLA;PLA;LDA@8;STA#321;LDA@0;STA#33C;STA#357;STA#372
  130 JMP#C55B;]
  140 N.;E.
```

>> PROGRAMMA "ZERO"-FUNCTIE

```
>L.
   10 REM ZERO
   11 DIM LL2
   12 IN."BASE ADRESS"Q
   20 F.X=0  TO 1;LLX=-1;N.
   30 F. X=0 TO 1
   40 P=Q
   50[LDY@26;:LL1LDA@0;STA#322,Y;STA#33D,Y;STA#358,Y;STA#373,Y
   60 DEY;BPLLL1;JMP#C55B;]
   70 N.;E.


>RUN
```

HANS VAN DER LINT
ROOSENDAAN STRAAT 26
OUD GASTEL   tel 01651-1553

S H U T T...... Een nieuw statement.

De Josbox-bezitters onder ons hebben een extra arsenaal van statements ter beschikking, die een nuttige aanvulling zijn op de al beschikbare statements van de Acorn Atom.
Toch blijven er nog handige statements over, die vanwege ruimtegebrek niet in de Josbox pasten.

Wat zou U bv. denken van een opdracht die alle variabelen in één keer op nul zet, zoals in de Toolkit van Telec.
Dit is vrij eenvoudig te realiseren met behulp van een statement die de Acorn al van nature kent, maar wat genegeerd wordt, als er geen diskdrive is aangesloten, namelijk: "SHUT".
Als we "SHUT" gevolgd door een variabele intikken, wordt er gesprongen naar de R.A.M. locaties #21A en #21B.
Op die locaties staat het R.O.M. adres #C287, waar de computer vervolgens naar toe gaat. Op dit adres staat een RTS opdracht.
De computer doet dus.... Niets! Dat is natuurlijk zonde.

Als we nu het adres op de locaties #21A en #21B veranderen in een R.A.M. adres, dan kunnen we "SHUT" voor ons karretje spannen, en de integervariabelen op nul laten zetten. Bv:

```
10 P=#2800                   ;REM  Begin machinecode.
20[LDA@0;LDX@149             ;REM  Initialiseren.
30 STA 653,X;INX             ;REM  Nul in geheugen, X=X+1
40 BNE#2804;RTS;]            ;REM  Klaar?
50 !#21A=#2800;END           ;REM  Vector goed zetten.
```

Als we nu "SHUT" gevolgd door een willekeurige variabele intikken, (Het simpelste is gewoon "SHUTT"), worden de Zero-page adressen #322....#38C vol met nullen gezet.
Dit heeft als gevolg dat alle integervariabelen de waarde nul krijgen.
Het statement "SHUTT" werkt zowel in de direct mode, als in programma's. Bovenstaand programma mag dus gewoon overgeschreven worden door een ander programma.
Na een BREAK moet U het statement heraktiveren met: !#21A=#2800
Succes!

Frank de Groot